



**HAL**  
open science

# Spatiotemporal description and modeling of mechanical product and its assembly sequence based on mereotopology : theory, model and approach

Elise Gruhier

## ► To cite this version:

Elise Gruhier. Spatiotemporal description and modeling of mechanical product and its assembly sequence based on mereotopology : theory, model and approach. Other. Université de Technologie de Belfort-Montbéliard, 2015. English. NNT : 2015BELF0276 . tel-01878596

**HAL Id: tel-01878596**

**<https://theses.hal.science/tel-01878596>**

Submitted on 21 Sep 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SPIM

## Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques

UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

# Spatiotemporal description and modeling of mechanical product and its assembly sequence based on mereotopology: Theory, model and approach

Description et modélisation spatio-temporelle du couple produit-process d'assemblage basées sur la méréotopologie : théorie, modèle et approche

■ ELISE GRUHIER



# SPIM

## Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques  
UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

N° X | X | X

THÈSE présentée par

**ELISE GRUHIER**

pour obtenir le

Grade de Docteur de

l'Université de Technologie de Belfort-Montbéliard

Spécialité : **Mécanique**

### Spatiotemporal description and modeling of mechanical product and its assembly sequence based on mereotopology: Theory, model and approach

Description et modélisation spatio-temporelle du couple produit-process  
d'assemblage basées sur la méréotopologie : théorie, modèle et approche

Unité de Recherche :  
IRTES-M3M

Soutenue publiquement le 4 décembre 2015 devant le Jury composé de :

PROF. MICHEL TOLLENAERE	Président	Institut National Polytechnique de Grenoble
PROF. ALAIN BERNARD	Rapporteur	Ecole centrale de Nantes
PROF. PHILIPPE VERON	Rapporteur	Arts et Métiers ParisTech – Aix-en-Provence
PROF. IMRE HORVATH	Examineur	Delft University of Technology (The Netherlands)
DR. KYOUNG-YUN KIM	Examineur	Wayne State University (USA)
PROF. SAMUEL GOMES	Directeur de thèse	Université de Technologie de Belfort-Montbéliard
PROF. SAID ABOUDI	Co-directeur de thèse	Université de Technologie de Belfort-Montbéliard
DR. FRÉDÉRIC DEMOLY	Encadrant	Université de Technologie de Belfort-Montbéliard





# Acknowledgments

First, I would like to thank Prof. Alain BERNARD and Prof. Philippe VERON from the advising committee members, who have accepted to evaluate my research work. I also would like to acknowledge Prof. Imre HORVATH and Dr. Kyoung-Yun KIM, other members of the examination committee, who gave me the honor to evaluate my research works.

I would like to thank Prof. Michel TOLLENAERE to have presided over the jury of the PhD defense.

I would like to express my sincere gratitude towards my supervisor, Prof. Samuel GOMES, who has trusted me to achieve this PhD work and also during the lectures for engineering students and the industrial projects.

I also would like to express my gratefulness to my co-supervisor Prof. Said ABBOUDI for his advices.

I would like to express my highest gratitude to my instructor, Dr. Frédéric DEMOLY for his guidance throughout my PhD, his patience and his precious advices and comments concluding our numerous discussions.

I am also thankful for the UTBM welcome in the INCIS staff of the IRTES-M3M laboratory.

I would like to acknowledge Olivier DUTARTRE, the IT engineer, for his feedbacks concerning the IT side of my PhD.

I would like to thank Catherine EBERSTEIN, Béatrice ROSSEZ and Pascal ALDINGER, who help me to feel well integrated within the laboratory. Even if I did not work directly with them, my appreciation is also extended to all my other colleagues for their welcoming and their good mood.

Lastly, special thanks to my family - my parents, Anne-Laure and Robin - for their support, love and constant encouragements throughout my PhD studies.



# Contents

<b>Acknowledgments</b>	<b>i</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of acronyms</b>	<b>xv</b>
<b>General introduction</b>	<b>xvii</b>
<b>Chapter 1 Context and problem statement</b>	<b>1</b>
1.1 Industrial context . . . . .	2
1.2 Scientific context . . . . .	4
1.3 Problem statement and research aims . . . . .	6
1.3.1 Problem statement . . . . .	6
1.3.2 Research aims . . . . .	7
1.4 Summary . . . . .	9
<b>Chapter 2 State of the art in the domain of mereotopological theories, ontology and product lifecycle management</b>	<b>11</b>
2.1 Introduction . . . . .	13
2.2 Mereotopology-based theories . . . . .	13
2.2.1 Definition . . . . .	13
2.2.1.1 Mereology . . . . .	13
2.2.1.2 Mereotopology versus mereogeometry . . . . .	14
2.2.1.3 Four-dimensionalism . . . . .	15
2.2.2 Background in Mereotopology . . . . .	16
2.2.2.1 Common core of mereotopology . . . . .	16

2.2.2.2	Description and classification of the mereotopology-based theories . . . . .	17
2.2.3	Main approaches in product design . . . . .	22
2.2.3.1	Salustri's approach . . . . .	22
2.2.3.2	Approach from TU Delft . . . . .	22
2.2.3.3	Kim's approach . . . . .	24
2.2.3.4	Demoly's approach . . . . .	24
2.2.4	Remaining challenges and emerging needs . . . . .	26
2.2.4.1	Basic problems in mereotopology . . . . .	26
2.2.4.2	Assembly-Oriented Design . . . . .	27
2.2.4.3	Related issues and needs in engineering design and CAD knowledge representation . . . . .	29
2.2.5	Summary and position . . . . .	29
2.3	Ontology models . . . . .	31
2.3.1	Ontology definition and goal . . . . .	31
2.3.2	Product design rationale, intents and formalisms . . . . .	32
2.3.3	Spatiotemporal ontologies . . . . .	33
2.3.4	Main ontology models in product design . . . . .	35
2.3.4.1	FBS model . . . . .	35
2.3.4.2	PRONOIA approach . . . . .	36
2.3.4.3	CPM and OAM models . . . . .	36
2.3.4.4	AsD approach . . . . .	39
2.3.5	Summary and position . . . . .	40
2.4	Product Lifecycle Management . . . . .	41
2.4.1	PLM definition and goal . . . . .	41
2.4.2	Engineering change management . . . . .	42
2.4.2.1	Changes . . . . .	42
2.4.2.2	Definition and needs of ECM . . . . .	44
2.4.3	Main engineering approaches to maintain consistency in product design and BIM . . . . .	45
2.4.3.1	Demoly's et al. approach . . . . .	45
2.4.3.2	Louhichi's and Rivest's approach . . . . .	45
2.4.3.3	Chen's and Luo's approach . . . . .	45
2.4.4	Heterogeneity management in other domains . . . . .	46

---

2.4.5	Summary and position . . . . .	48
2.5	Summary: need of theory, model and approach . . . . .	48
<b>Chapter 3 Spatiotemporal mereotopology-based theory for product-process description – JANUS</b>		<b>51</b>
3.1	Introduction . . . . .	52
3.2	Overall description of JANUS theory . . . . .	53
3.2.1	General philosophy . . . . .	53
3.2.2	Object change description . . . . .	54
3.3	Description of the spatial dimension of JANUS . . . . .	55
3.4	Description of the temporal description of JANUS . . . . .	57
3.5	Description of the spatiotemporal description of JANUS . . . . .	58
3.5.1	Basic problems with spatiotemporal visualization . . . . .	58
3.5.2	Filiation relationships . . . . .	61
3.5.3	Spatiotemporal primitives classification . . . . .	61
3.5.4	Kinematic pairs and Design changes description . . . . .	62
3.5.5	Technological pairs description . . . . .	64
3.6	Summary . . . . .	76
<b>Chapter 4 Development and implementation of the formal ontology – PRONOIA2</b>		<b>77</b>
4.1	Introduction . . . . .	78
4.2	Overview of the research approach . . . . .	78
4.3	Design rationale . . . . .	82
4.4	Meta-ontology description . . . . .	82
4.5	Assembly-ontology description . . . . .	84
4.6	OWL-DL restriction . . . . .	88
4.6.1	General information on OWL-DL . . . . .	88
4.6.2	Restriction in the meta- and assembly-ontology . . . . .	90
4.7	SWRL rules . . . . .	95
4.8	Summary . . . . .	97
<b>Chapter 5 A PLM-based approach to manage assembly and design evolutions – MERCURY</b>		<b>99</b>
5.1	Introduction . . . . .	100



5.2	Approach objectives . . . . .	100
5.3	Two different applications: Assembly and design evolutions . . . . .	103
5.3.1	Product evolution control at the early design stages . . . . .	103
5.3.2	Design evolution control during the design process . . . . .	104
5.4	Management of product-process information consistency . . . . .	104
5.5	Overall framework of MERCURY . . . . .	110
5.6	Overview of information flow through PLM . . . . .	113
5.7	Summary . . . . .	122
<b>Chapter 6 Case studies illustrating the theory, model and approach</b>		<b>123</b>
6.1	Introduction . . . . .	124
6.2	First case study: Simplified academic case study . . . . .	124
6.2.1	Mereotopological description of the case study . . . . .	124
6.2.2	Swept volumes representation . . . . .	125
6.3	Second case study: Design of a medical drone . . . . .	129
6.3.1	Description of the case study . . . . .	129
6.3.2	Case study solving following the approach on assembly evolution . . . . .	129
6.3.3	Populating PRONOIA2 . . . . .	134
6.3.4	Reasoning - consistency checking . . . . .	135
6.3.5	Case study solving following the approach on design evolution . . . . .	136
6.4	Summary . . . . .	142
<b>Chapter 7 Conclusions and future work</b>		<b>147</b>
7.1	Overall conclusion . . . . .	148
7.2	Future work . . . . .	149
7.2.1	Short-term future work: CAD support improving designer's understanding . . . . .	149
7.2.2	Mid-term future work: Transformable products description . . . . .	150
7.2.3	Long-term future work: Consideration of other lifecycle stages and domains . . . . .	151
<b>References</b>		<b>153</b>
1	Research work publications . . . . .	153
1.1	Peer-reviewed international journals . . . . .	153
1.2	Peer-reviewed international conferences . . . . .	153

---

1.3	Peer-reviewed national symposium . . . . .	154
2	Other references . . . . .	154
		<b>154</b>



# List of Figures

1	Representation of the thesis structure . . . . .	xix
2	Proposed vision of the research . . . . .	8
3	Tree of investigated research fields [Gruhier et al., 2014a] . . . . .	13
4	Representation of the “transitive problem” . . . . .	14
5	Representation of the Smith’s mereotopological primitives [Smith, 1996] . .	17
6	The eight RCC8 relations [Randell et al., 2012] . . . . .	18
7	The six motions classes of Muller [Muller, 1998] . . . . .	19
8	Visualization of Stell and West’s operators [Stell and West, 2004] . . . . .	20
9	Nucleus-based U-P-E model [Van der Vegte and Horvath, 2003] . . . . .	23
10	Exemple of fusion welding [Kim et al., 2008] . . . . .	24
11	Mereotopological representations of some assembly joints [Kim et al., 2008]	25
12	Representation of temporal relations for regions $r$ , $u$ , $v$ and $w$ [Demoly et al., 2012b] . . . . .	26
13	Map of existing mereotopological theories in the domain of assembly-oriented design . . . . .	30
14	Layer approach of the Semantic Web [Antoniou and Van Harmelen, 2008] .	31
15	Hierarchy of the SUMO ontology [Pease, 2014] . . . . .	33
16	FBS model under the viewpoint of the engineering design knowledge [Colombo et al., 2007] . . . . .	35
17	The PRONOIA ontology model in Protégé [Demoly et al., 2012b] . . . . .	36
18	Core Product Model, OWL version [Fiorentini et al., 2010] . . . . .	38
19	Open Assembly Model, OWL version [Fiorentini et al., 2010] . . . . .	38
20	Assembly Design ontology class hierarchy [Kim et al., 2008] . . . . .	39
21	Map of existing ontologies along lifecycle phases [Gruhier et al., 2015a] . .	41
22	Definition process of a minimal skeleton model [Demoly et al., 2011a] . . .	46
23	Determination of association between CAD elements [Louhichi and Rivest, 2014] . . . . .	47
24	Relationship between construction activities and BIM components [Chen and Luo, 2014] . . . . .	47
25	Map of existing management approaches . . . . .	49
26	Introduction of a semantics and logics layer for ensuring product-process integration . . . . .	52

27	The three temporal situations of Bergson applied to assembly design on which the proposed theory is based [Gruhier et al., 2014e] . . . . .	53
28	Geometric representation of $k$ and $f$ entities . . . . .	56
29	Example of cylindrical pair assembly and related skeletons (i.e. assembly and interface skeletons) . . . . .	57
30	Decomposition of <i>Part 5</i> insertion [Gruhier et al., 2015b] . . . . .	60
31	Description of <i>CylindricalOP</i> in a spatiotemporal graph [Gruhier et al., 2015b] . . . . .	67
32	Cartography of the evolutions to be considered in PRONOIA2 [Gruhier et al., 2015c] . . . . .	68
33	Cartography of the technological pairs to be considered in PRONOIA2 . . . . .	76
34	PRONOIA2 ontology model built from JANUS theory . . . . .	78
36	Proposed framework of the PRONOIA2 approach . . . . .	80
35	Mereotopological product relationships description approach (PRONOIA2) over space and time . . . . .	81
37	Overall view of PRONOIA2 ontology . . . . .	83
38	Classes, sub-classes and properties of the meta-ontology (PRONOIA2) [Gruhier et al., 2014d] . . . . .	84
39	Classes and sub-classes related to spatial regions of the domain-ontology (PRONOIA2) [Gruhier et al., 2015a] . . . . .	85
40	Classes and sub-classes related to temporal regions of the domain-ontology (PRONOIA2) [Gruhier et al., 2015a] . . . . .	85
41	Classes and sub-classes related to spatiotemporal regions of the domain-ontology (PRONOIA2) . . . . .	86
42	Classes and sub-classes related to spatial primitives of the domain-ontology (PRONOIA2) [Gruhier et al., 2015a] . . . . .	86
43	Classes and sub-classes related to temporal primitives of the domain-ontology (PRONOIA2) [Gruhier et al., 2015a] . . . . .	87
44	Classes and sub-classes related to spatiotemporal primitives of the domain-ontology (PRONOIA2) [Gruhier et al., 2015a] . . . . .	87
45	Building of the domain-ontology . . . . .	88
46	Classes, sub-classes and properties of the domain-ontology (PRONOIA2) [Gruhier et al., 2014d] . . . . .	88
47	Result of consistency checking of PRONOIA2 within Protégé using Hermit 1.3.7 . . . . .	90
48	Management approach based on PRONOIA2 ontology and JANUS theory	101
49	How to orchestrate information flow between PDM, MPM and CAD? . . . . .	102
50	Lifecycle positions of both applications . . . . .	103
51	Mechanical assembly example for both applications . . . . .	103
52	Representation of product evolution in the AOD context . . . . .	104
53	Representation of design changes . . . . .	105
54	Research map of PLM systems and CAD tools . . . . .	106

---

55	Link between spatial and temporal information through spatiotemporal relationships . . . . .	106
56	Reconciliation between PDM and MPM systems . . . . .	107
57	Comparison between the information flow in the MUVOA model [Demoly et al., 2012d] and the proposed model . . . . .	109
58	Simplified framework to manage product-process information continuity . .	110
59	Proposed framework to manage product-process information continuity in the context of AOD . . . . .	112
60	Proposed approach to bridge the gap of heterogeneity . . . . .	113
61	Evolution of PLM systems and CAD application implementation . . . . .	114
62	Type of information travelling through PLM systems and CAD application	115
63	Proposed implementation of PRONOA2 ontology within the hub . . . . .	118
64	Information flow centered on PRONOA2 ontology for the assembly evolution application . . . . .	120
65	Information progress through the product development process . . . . .	121
66	Graph of part-to-part relationships of the first case study . . . . .	124
67	Description of the whole assembly process in a spatiotemporal graph for designers – first case study . . . . .	125
68	Spatial representation of the whole assembly process for the first case study	126
69	CAD representation of the whole assembly within CATIA v5 for the first case study . . . . .	127
70	Design evolution of the first case study step by step . . . . .	127
71	Representation of a pre-concept using morphogenesis for the case study “Medical drone” . . . . .	130
72	eBOM structure within MPM system (Notixia) for the “medical drone” case study . . . . .	130
73	Graph of part-to-part relationships of the second case study . . . . .	131
74	Assembly operations representation within MPM system (Notixia) for the “medical drone” case study . . . . .	132
75	Representation of the assembly sequence planning of the second case study within MPM system (Notixia) . . . . .	133
76	Restructured eBOM of the second case study within CAD application (Catia)	134
77	Description of the whole assembly process in a spatiotemporal graph for the second case study . . . . .	135
78	Spatial representation of the whole assembly process for the second case study . . . . .	136
79	Instances of the ontology with the “kinematic pairs positioning” primitives for the second case study . . . . .	137
80	Result of consistency checking of the second case study within Protégé . .	138
81	Query “Show Swept Volume” and answers for the second case study . . . .	138
82	Query “Show all swept volumes in the second case study and their related spatiotemporal primitives” and answers . . . . .	139
83	Instances of the ontology with the “design changes” primitives for the second case study . . . . .	140



84	Description of the changes undertaken during the design process in a spatiotemporal graph for the second case study . . . . .	140
85	UML diagram for the second case study showing activities and the related stakeholders during the design process . . . . .	141

# List of Tables

1	Comparison between endurantism- and perdurantism-based approaches [Al-Debei et al., 2012] . . . . .	15
2	Fundamental mereotopological operators [Smith, 1996] . . . . .	17
3	RCC8 mereotopological primitives description . . . . .	18
4	Classification table of relevant region-based theories [Gruhier et al., 2014b]	21
5	Mereotopological description of kinematic pairs Mereotopological description of kinematic pairs [Demoly et al., 2012b] . . . . .	26
6	Some of the description logic restrictions of PRONOIA [Demoly et al., 2012b]	37
7	Some of the SWRL rules [Kim et al., 2008] . . . . .	40
8	Spatial mereotopological primitives description . . . . .	55
9	Temporal mereotopological primitives description and representation . . .	59
10	Assembly types of three components . . . . .	60
11	Part-to-part relationships definition for spatiotemporal primitives [Demoly et al., 2012b] . . . . .	62
12	Classification of changes and associated spatiotemporal primitives . . . . .	63
13	Mereotopological description of the “kinematic pairs positioning” primitives [Gruhier et al., 2015b] . . . . .	65
14	Mereotopological description of the “design changes” primitives [Gruhier et al., 2015b] . . . . .	66
15	Legend of the spatiotemporal graph . . . . .	68
16	Representation of the spatiotemporal kinematic pairs (1) . . . . .	69
17	Representation of the spatiotemporal kinematic pairs (2) . . . . .	70
18	Classification of technological assembly operations . . . . .	70
19	Mereotopological description of non-permanent assembly with added entities [Gruhier et al., 2015b] . . . . .	71
20	Mereotopological description of permanent assembly with added entities (1) [Gruhier et al., 2015b] . . . . .	72
21	Mereotopological description of permanent assembly with added entities (2) [Gruhier et al., 2015b] . . . . .	73
22	Mereotopological description of permanent assembly with added entities (3) [Gruhier et al., 2015b] . . . . .	74
23	Mereotopological description of permanent assembly without added entities [Gruhier et al., 2015b] . . . . .	75

---

24	PRONOA2 DL rules in the meta-ontology . . . . .	90
25	PRONOA2 DL rules in the assembly-ontology . . . . .	91
26	PRONOA2 DL rules in the assembly-ontology . . . . .	92
27	PRONOA2 DL rules in the assembly-ontology . . . . .	93
28	PRONOA2 DL rules in the assembly-ontology . . . . .	94
29	SWRL rules in PRONOA2 ontology . . . . .	96
30	Changes management and consideration . . . . .	102
31	Link between stakeholders and views in product and assembly process . . .	109
32	Functions realised in Pegasus and their related stakeholders . . . . .	117
33	Information available in the different information systems and applications	119
34	Functionalities and concepts of the hub . . . . .	119
36	Parts list of the first case study . . . . .	125
35	Mereotopological description of the first mechanical case study . . . . .	128
37	Functions and related solutions for the second case study . . . . .	129
38	Temporal information for the second case study: assembly operations and their associated primitives . . . . .	132
42	Parts list of the second case study . . . . .	141
39	Mereotopological description of the second mechanical case study (1) . . .	143
40	Mereotopological description of the second mechanical case study (2) . . .	144
41	Second case study representation . . . . .	145

# List of acronyms

<b>ACSP</b>	<b>A</b> telier <b>C</b> oopératif de <b>S</b> uivi de <b>P</b> rojet (in french)
<b>AOD</b>	<b>A</b> ssembly- <b>O</b> riented <b>D</b> esign
<b>BOL</b>	<b>B</b> eginning <b>O</b> f <b>L</b> ife
<b>CAD</b>	<b>C</b> omputer- <b>A</b> ided <b>D</b> esign
<b>DL</b>	<b>D</b> escription <b>L</b> ogic
<b>eBOM</b>	engineering <b>B</b> ill <b>O</b> f <b>M</b> aterials
<b>EOL</b>	<b>E</b> nd <b>O</b> f <b>L</b> ife
<b>GIS</b>	<b>G</b> eographic <b>I</b> nformation <b>S</b> ystem
<b>JANUS</b>	<b>J</b> oined <b>A</b> ware <b>N</b> ess and <b>U</b> nderstanding in assembly-oriented de <b>S</b> ign with mereotopology
<b>mBOM</b>	<b>m</b> anufacturing <b>B</b> ill <b>O</b> f <b>M</b> aterials
<b>MERCURY</b>	a <b>M</b> anag <b>E</b> ment app <b>R</b> oa <b>C</b> h of prod <b>U</b> ct and process <b>R</b> elationships in assembly and design phases
<b>MOL</b>	<b>M</b> iddle <b>O</b> f <b>L</b> ife
<b>MPM</b>	<b>M</b> anufacturing <b>P</b> rocess <b>M</b> anagement
<b>MUVOA</b>	<b>M</b> ultiple <b>V</b> iewpoints <b>O</b> riented <b>A</b> ssembly
<b>OWL</b>	<b>O</b> ntology <b>W</b> eb <b>L</b> anguage
<b>PDM</b>	<b>P</b> roduct <b>D</b> ata <b>M</b> anagement
<b>Pegasus</b>	<b>P</b> roduct design <b>E</b> ngineering based on <b>G</b> enerative <b>A</b> ssembly <b>S</b> eq <b>U</b> ence <b>S</b> planning
<b>PLM</b>	<b>P</b> roduct <b>L</b> ifecycle <b>M</b> anagement
<b>PRONOIA</b>	<b>P</b> ROduct relatio <b>N</b> ships description based <b>O</b> n mereotopolog <b>I</b> c <b>A</b> l theory
<b>RDF</b>	<b>R</b> esource <b>D</b> escription <b>F</b> ramework
<b>SRF</b>	<b>S</b> patial <b>R</b> egion – <b>F</b> inal
<b>STR</b>	<b>S</b> patio <b>T</b> emporal <b>R</b> egion
<b>SWRL</b>	<b>S</b> emantic <b>W</b> eb <b>R</b> ule <b>L</b> anguage
<b>TR</b>	<b>T</b> emporal <b>R</b> egion



# General introduction

Over the last two decades, the industrial competitive context and research advances have been led to consider new competitiveness drivers so as to ensure high-volume production and decrease product lifecycles. In the context of global knowledge-based economy, it has been widely accepted that **Information Communication Technology (ICT)** provides key drivers to design and manufacturing innovation, efficiency and productivity [Riemenschneider, 2014]. Products need to be more customizable and oriented towards the whole lifecycle. Although ICT have provided fruitful results into the world of industrial automation, IT solutions are currently developed on islands without an integrated overall architecture design and without exploiting its full potential. Indeed, due to the complex nature of product delivery processes (especially design and manufacturing) in industry and an obvious lack of models, methodologies and tools, further research effort is highly recommended with respect to:

- an appropriate integration of lifecycle knowledge in product design;
- a better integration of product development with its management process, including enhanced dynamic product models;
- a better alignment of virtual and physical aspects of product lifecycles;
- a fast knowledge creation and exchange process.

Hence, the impact of rapid developments in ICT is currently accepted as a major issue to innovation in both products and processes, such as mixed technology products (e.g. mechatronic/infotronic/transformable products) and interdisciplinary and cross-enterprise approaches to collaboration in product development and manufacturing. Product design is understood as a major activity in the product lifecycle that has to take into consideration various product lifecycle issues such as manufacturing capabilities. This led to new research challenges to be tackled, especially new design methodologies ensuring proactivity and knowledge reuse, new design platform with embedded intelligence, new design model linking virtual and physical aspects, etc.

In such a context, new organization mode such as concurrent engineering has emerged to replace sequential mode [Koufteros et al., 2001]. Concurrent engineering requires a high communication level between all project stakeholders [Eynard, 2005] and an efficient information flow. Few years later, **Product Lifecycle Management (PLM)** strategies have



arisen so as to share product data among processes and organizations [PLM IWG, 2007].

The increasing product complexity and reducing development time require to develop new approaches, so as to aid product architects and designers in their activities. Indeed, due to 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. As such, 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, in ontological applications, knowledge-based approaches, models and so on.

New paradigms, coupling product modeling with semantics, are spreading in the research community. This paradigm enables more interoperability between design and assembly processes. This research work is focused on the **Assembly-Oriented Design (AOD)** context, in which **DFA (Design For Assembly)** analyses and assembly sequence generation are integrated and product structure is manipulated. Hence, design and assembly phases are integrated at the early design stages and the different processes are overlapping [Valle and Vazquez-Bustelo, 2009].

In the current context, design and assembly phases are not integrated in a seamless manner. Actually, current product modeling is only considered from a geometric point of view, but its representation in real world is not yet considered in an appropriate manner. As such, knowledge about product and its assembly sequence should be formalized in a manner close to the reality. In fact, this formalization enables capturing the product story, which can be used for traceability or for prediction in future developments. A similar challenge already exists in the domain of **Internet of Thing (IoT)** [Weber and Weber, 2010]. Its purpose is to overcome the gap between objects in the physical world and their representation in information systems. A definition given by Haller [Haller et al., 2009] is: “a world where physical objects are seamlessly integrated into the information network and where the physical objects can become active participants in business processes”. Hence, with the consideration of this challenge, the product evolution in design and assembly process will be understood.

The PhD research works have been done within the **INCIS** (in french: **Ingénierie Numérique avancée pour la Conception Intégrée de Systèmes mécaniques**) research team, led by Prof. Samuel Gomes in the **IRTES-M3M** laboratory, in french: **Institut de Recherche sur les Transports, l’Energie et la Société - Mécatronique, Méthodes, Modèles et Métiers (EA 7274)**. This team focus its research works on the management of product lifecycle, founded on knowledge-based engineering [Gomes, 2008] [Monticolo, 2008]. Research works have been particularly carried out on the improvement of routine design processes in the early design stages. As such, a PLM platform – **ACSP** (in french: **Atelier Coopératif de Suivi de Projet**) – has been developed, as well as a hub – called **Pegasus (Product design Engineering based on Generative Assembly SeqUenceS plan-**

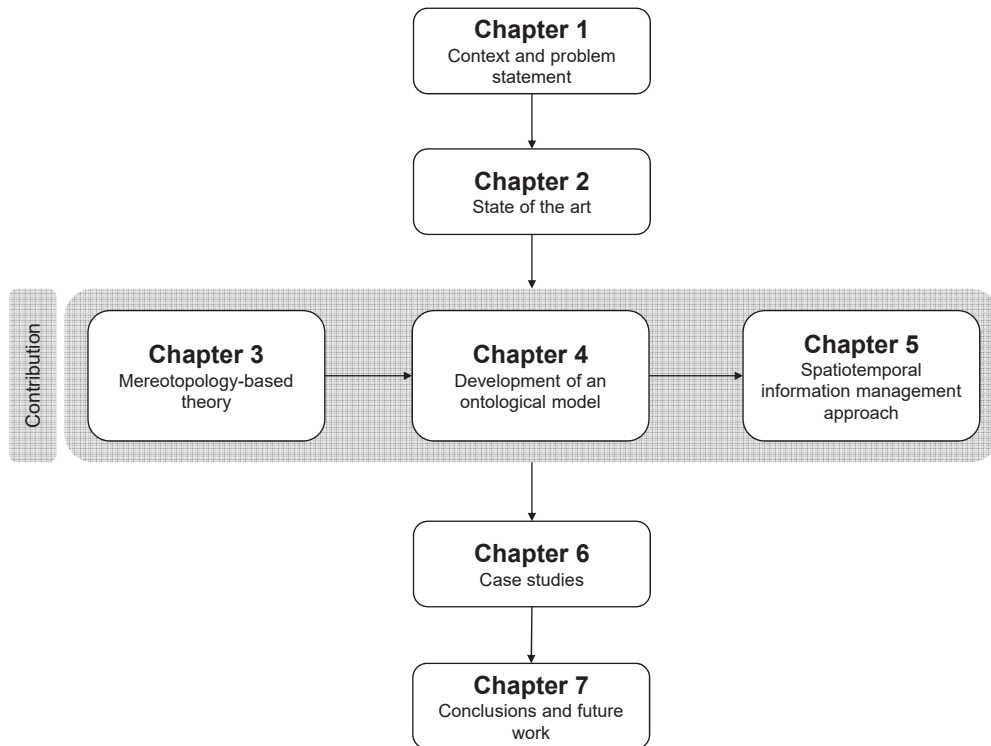


Figure 1: Representation of the thesis structure

ning) – to orchestrate product-process information [Demoly, 2010].

The thesis summarizes our research works on proactive engineering and information management in PLM systems. Here the objective is to formally describe product-process information and especially spatiotemporal relationships is proposed, so as to enable the understanding of products evolution and improve product architects’ and designers’ awareness. A mereotopological theory and its related ontology have been developed, in order to achieve our objective. As such, the thesis is composed of seven chapters structured as on Figure 1. The scientific contribution is presented in chapters 3, 4 and 5.

The **first chapter**, entitled “*Context and problem statement*”, introduces the industrial stakes and scientific context, thus defining the problem statement limits of our research works.

The **second chapter**, entitled “*State of the art in the domain of mereotopological theories, ontology and product lifecycle management*”, presents a comprehensive literature review on mereotopology-based theories, spatiotemporal ontologies and information management approach to highlight the research issue.

The **third chapter**, entitled “*Spatiotemporal mereotopology-based theory for product-process description - **JANUS** (**J**oined **A**ware**N**ess and **U**nderstanding in assembly-oriented de**S**ign with mereotopology)*”, describes the proposed theory, which provides product-process associations with mereotopology in three dimensions (i.e. spatial, temporal and spatiotemporal). The theory enables the description of product evolution through its early product development stages.

The **chapter four**, entitled “*Development and implementation of the formal ontology - **PRONOIA2** (**P**ROduct relatio**N**ships description based **O**n mereotopolog**IcAl** theory **2**)*”, introduces an ontological implementation of the formal theory through semantics and logics within Protégé and **OWL** (**O**ntology **W**eb **L**anguage). Moreover, rules are introduced within the ontology with **DL** (**D**escription of **L**ogic) and **SWRL** (**S**emantic **W**eb **R**ule **L**anguage) in order to check information consistency.

The **chapter five**, entitled “*A PLM-based approach to manage assembly and design evolutions - **MERCURY** (a **M**anag**E**ment app**R**oa**C**h of prod**U**ct and process **R**elationships in assembly and design phases)*”, presents a novel approach with its related framework and information flow (between information systems and ontology) to design a consistent product and improve product architects’ and designers’ understanding.

The **chapter six**, entitled “*Case studies illustrating the theory, model and approach*”, presents two mechanical assemblies to illustrate the relevance of the developed research work.

The **chapter seven**, entitled “*Conclusions and future work*”, draws the main conclusions and introduces the future work to extend the actual approach in other product lifecycle stages or domains and adapt it to other kind of products (e.g. transformable product).

# Chapter 1

## Context and problem statement

### Contents

---

<b>1.1</b>	<b>Industrial context . . . . .</b>	<b>2</b>
<b>1.2</b>	<b>Scientific context . . . . .</b>	<b>4</b>
<b>1.3</b>	<b>Problem statement and research aims . . . . .</b>	<b>6</b>
1.3.1	Problem statement . . . . .	6
1.3.2	Research aims . . . . .	7
<b>1.4</b>	<b>Summary . . . . .</b>	<b>9</b>

---

This chapter describes the industrial stakes and the scientific context, on which the problem statement will be based and formulated.

## 1.1 Industrial context

The current industrial context is faced to severe competition and leads companies to make numerous tradeoffs in terms of internal and external development strategies. Industry is at a turning point where production strategy (i.e. global versus local), product strategy (e.g. short versus long-life cycle), information/knowledge capture/accessibility strategy (e.g. ontology-based, cloud-based, etc.), to name a few, need to be addressed in a well-balanced manner. From an internal point of view, engineering processes become more and more knowledge-intensive and then require adapted intelligent environment, especially in product design and manufacturing. Indeed during the product development process, product architects and designers decisions have a major impact on downstream processes and then demand better understanding and awareness in their activities. Currently, the knowledge-intensive and scattered information systems covering the beginning of product lifecycle also requires advanced interaction, information flows consistency and interpretable description, especially in the early design phases where proactive integration of knowledge from downstream processes and qualitative specification are needed [Demoly et al., 2013b].

Current PLM systems enable the management of information in the **B**eginning **O**f **L**ife (**BOL**), **M**iddle **O**f **L**ife (**MOL**) and **E**nd **O**f **L**ife (**EOL**) phases [Garetti, 2013]. BOL includes design and manufacturing phases, which are critical steps of the product development process, especially the early design stages [Salustri et al., 2008], where the geometry is not yet defined. At this stage, designers do not have access to product architect's intents as early and fast as it should be in a collaborative project. Despite of the early design stages importance, supports for designers before embodiment stages are still insufficient. Besides, PLM has been specifically developed to aid project manager in their tasks. However, PLM fails to concretely aid designers in their activities, which leads to a lack of understanding along product design stages. For instance, Ryerson University has highlighted through a survey that **S**mall- and **M**edium-**S**ized **E**nterprises (**SMEs**) are lacking support for conceptual design [Salustri et al., 2008]. Besides, Robinson [Robinson, 2012] states that 20 % of designers' activities is dedicated to understand the information. Therefore additional supports and assistance are required, especially in the early design stages. Information also needs to be made more accessible and understandable. This can lead to the formalization of information [Levenchuck, 2012], in order to overcome design mistakes and misunderstanding due to a lack of formal information.

The PLM chain is composed of **P**roduct **D**ata **M**anagement (**PDM**) and **M**anufacturing **P**rocess **M**anagement (**MPM**) systems in engineering and manufacturing. PLM systems also manage spatial and temporal information [Peachavanish et al., 2006] within complex engineering projects. On one hand, PDM manages spatial information such as engineering **B**ill **O**f **M**aterials (**eBOM**), part, product structure, documents and so on.

On the other hand, MPM manages temporal information such as **manufacturing Bill Of Materials (mBOM)**, assembly operation and assembly sequence and plan to name a few. Although PDM and MPM are considered as heterogeneous systems, PLM systems do not bridge enough the gap between product and process models [Swain et al., 2014]. Based on these statements, a lack of associativity between technical objects in PLM is highlighted [Demoly et al., 2012b].

As such, PLM suffers from non-efficient information flow, which does not enable concurrent engineering [Helms, 2002] between product design and assembly phases. Indeed, a better collaboration between product and process is required [CIMdata, 2015]. The future of PLM lies in new approaches with semantics [Filos, 2012]. Indeed, current PLM systems need to:

- Use formal language to decrease the number of mistakes in product definition [Levenchuck, 2012];
- Enable a seamless integration of assembly sequence information in product design;
- Enable information interoperability;
- Capture information and meaning about changes description [MacKrell, 2015];
- Have common semantics between different stakeholders;
- Ensure the product-process information consistency between different information systems;
- Make design intents explicit to share information between product architects and designers.

Therefore, current industrial challenges consist in ensuring knowledge and information consistency [Pittet et al., 2014] throughout the product development process, from conceptual design phase to embodiment design phase. Consistency checking is critical in product modeling phase [Salustri, 2002] and requires a semantical and logical foundation in the context of integrated design and concurrent engineering. This highlights needs in semantic associations between engineering and manufacturing entities, and especially focuses on the management of relationships [Witherell et al., 2013]. Hence, product development approaches need to be rethink in order to provide consistent product-process information to stakeholders [Ross et al., 2008] and aid them carry out top-down product design more easily [Chen et al., 2010]. Here, product design requires to be described by proactively considering its assembly sequence as early as possible in the product development. Such description enables information consistency checking with preliminary information through PLM systems.



## 1.2 Scientific context

A current research challenge consists in the knowledge integration of products lifecycle phases (i.e. process planning, assembly planning, etc.) in an appropriate manner at the earliest product design stages [Demoly et al., 2012a]. To tackle this challenge, Demoly et al. [Demoly et al., 2012a] have highlighted the current stakes in engineering design and emerging needs in proactive engineering based on qualitative description of lifecycle knowledge, so as to improve designers' awareness and understanding. This statement has also been claimed by Kusiak and Salustri [Kusiak and Salustri, 2007], who have stated that design engineering is currently changing from an informal approach based on experience to a science-based approach [Zeng and Gu, 1999b].

Design is an evolving process that begins with design requirements and ends with product descriptions [Zeng and Gu, 1999a]. Despite of the encountered evolution, the dynamic aspect of the product definition has not been yet described in a seamless fashion [Zeng and Gu, 1999b]. Indeed, current **Computer-Aided Design (CAD)** applications are mainly dedicated to the spatial and geometric definition and do not capture the design story (that is quite different from a CSG tree for example) from a temporal point of view. Such design practices actually lead to static product definition and time-consuming efforts from manufacturing side in order to interpret assembly intents for example. This leads to difficulties in engineering definitions and wrong design interpretations. By following three-dimensionalism, design activity is mainly dedicated on the development of the spatial aspect of the product without considering intrinsically its evolution over time. A possible reason is due to the functional, geometric, physical complexity and multiple temporal configurations inferred in the design process, and also when the product information passes from design to manufacturing engineering and later stages [Chandrasegaran et al., 2013].

A similar issue can be identified in engineering change management, where changes are represented through objects' effectiveness. This results in the capture of multiple product configurations and increases complexity in the management of product design information. But evolution and changes (i.e. related to part, assembly, relationship, etc.) that may occur during the product definition are not adequately described, captured and represented. Indeed, changes need to be managed and tracked at the start of product lifecycle so as to control product evolution [MacKrell, 2015]. A formalization of these changes would provide a better awareness of product architects and designers. However, current PLM systems and CAD applications are not suited to capture and manage product evolution. Actually, current tools:

- Have a lack of product evolution and changes undergone recognition during the design and assembly phases;
- Provide a purely spatial definition of the product (through its structure and geometry);
- Do not capture information on the different states of the product evolution.

---

The fact of delivering product models in line with realistic situations over time can overcome these issues. Such stake opens new research challenges in engineering design, which covers different research areas, such as philosophical investigations, mathematics, artificial intelligence and engineering to name a few. In general, products models are considered as a representation of our perception of the reality from a three-dimensionalism (i.e. endurantism in philosophy) point of view. Literature has proved that researchers have built product models regarding their functionalities and their extension in space. A promising philosophy is about four dimensionalism or perdurantism, on which an object has distinct temporal parts throughout its existence. Here perdurantism considers the product as having the same identity whatever changes the product undergoes. With the addition of this philosophy, the model is considered as it is perceived in the real world. Perdurantism actually supports that objects have three spatial dimensions and move through time. In others words, this concept argues that material objects are persisting things wholly present at every moment of their existence. Based on these research stakes, new research efforts have to be addressed on formalism and theory in engineering design in order to represent product and lifecycle knowledge in a qualitative and machine-interpretable manner [Kusiak and Salustri, 2007].

Limits of current PLM approaches could be overcome by focusing on relationships, as well as considering spatiotemporal dimensions. Relational design is a design based on relationships, such as between parts and assembly operations. McKinney et al. [McKinney et al., 1996] state that representing relationships between time and space provides a powerful mechanism to communicate design intents. Each time a change in design occurs, relationships is changed. As such, stakeholders are aware of changes throughout the product development. However, a lack of consideration for spatiotemporal entities management is highlighted. Moreover, relational design is mainly a top-down approach, in which the whole product assembly is considered before the definition of its numerous parts. For instance, stakeholders will concentrate on the overall assembly (i.e. product) before designing parts. This approach enables framing the issue and then dividing it in sub-issues in order to facilitate the design process. With such an approach, product architects and designers have an overview of the product.

Our research works are based on national and international contributions, which are in harmony with us, especially in the domain of product-process description and information management approach in PLM. This enables our positioning towards the community. At the national level, the network **AIP-PRIMECA** (in french, **A**telier **I**nter-établissement de **P**roductique – **P**ôle de **R**essources **I**nformatiques pour la **M**ECANique) enables to reinforce scientific exchanges between PhD students and confirmed professors. Besides, **GDR-MACS** community (in french, **G**roupe **D**e **R**echerche du **C**NRS en **M**odélisation, **A**nalyse et **C**onduite des **S**ystèmes dynamiques) is focusing on management of product lifecycle process and of industrial knowledge. Several working groups are included in the community, especially **IS3C** (in french, **I**ngénierie des **S**ystèmes de **C**onception et **C**onduite du **C**ycle de vie produit) and **C2EI** (in french, **M**odélisation et pilotage des systèmes de **C**onnaissances et de **C**ompétences dans les **E**ntreprises **I**ndustrielles). At the international level, several communities are working on the same research fields than us.

One of them is the Design Society, which contributes to develop design activities and enables PhD students to present their works to renowned professors (especially during the SSEDR summer school). **PALM** (**P**roduct and **A**sset **L**ifecycle **M**anagement) has also the same objective of sharing experiences with young researchers. In addition, **IFIP** (**I**nternational **F**ederation of **I**nformation **P**rocessing) working group 5.7 contributes to research in the domain of “Advances in Production Management Systems”.

## 1.3 Problem statement and research aims

Previous sections have highlighted the industrial and scientific contexts in the domain of AOD. Problem statement and research aims will now be described in this section, so as to fulfil these expectations.

### 1.3.1 Problem statement

Based on previous industrial and scientific statements, research works have been focused on the improvement of product architect’s and designer’s awareness and understanding at the early design stages of product development. As such, the product evolution in the context of AOD needs to be described. Design engineering and manufacturing are currently changing their paradigm from an informal discipline based on experience to a domain based on science [Kusiak and Salustri, 2007].

Therefore, the fact that the “dynamic” aspect of design activity has not been yet taken into consideration in an appropriate manner is considered as a major issue [Zeng and Gu, 1999b]. Actually, the product definition evolves over time and changes along the design and even the assembly process. As such, the main issue is related to the description of the product and its evolutions in AOD. Such a description has to consider morphological features (i.e. physical and behavioral aspects [Rusak et al., 2004]) of the product, and capture design intents in order to get a product model close to the reality [Kim et al., 2006] and an easier interpretation for designers. In this context, the intentions can hardly be interpreted and lead to a higher number of revisions between the different stakeholders involved in the product lifecycle (especially between the design and assembly phases). The opportunity to add temporal and spatiotemporal aspects to the current research efforts is the key issue in order to be able to consider all geometric evolutions (e.g. deformation and transformation) and the move of products over time [Rusak et al., 2004].

Hence, the authors propose to develop their own theory, model and approach so as to promote a perdurantist vision [Al-Debei et al., 2012] in product design. Indeed, most of the research works have been based upon an endurantist point of view in product design so far, by only considering parts in the spatial dimension (i.e. 3D model). With such a metaphysic philosophy, changes in product design can be understood in terms of things, and this leads to limits if temporal objects from manufacturing have to be associated. To overcome this issue, a perdurantist vision in design has to be addressed in order to represent spatial, temporal and spatiotemporal objects in a unified manner. This therefore

enables the understanding of things in terms of changes. A strengthening of the scientific bases of design theories is required to formalize product-process definition. This formalization, by integrating logic and an appropriate language [Zeng and Gu, 1999b], enables the description of a strong “engineering sense” [Salustri, 2002].

As a consequence, the proposed research work is focused on the following issue: **“Spatiotemporal description and modeling of mechanical product and its assembly sequence based on mereotopology”**. Built on this, three research questions have been formulated as follows:

- How to describe the spatiotemporal evolution of the product in the context of assembly-oriented design?
- How to formally implement this description in order to be interpretable by PLM systems?
- How to integrate spatiotemporal description in the management of product-process information?

### 1.3.2 Research aims

Based on this problem statement, linked to industrial and scientific stakes previously described, the major contribution of this thesis can be defined as follows:

Propose a spatiotemporal description of product-process relationships, through a logical and semantical foundation in order to ensure a seamless integration of assembly sequence planning in product design

This issue will be addressed by using region-based theories in order to qualitatively describe the product, and create logic rules for product-process reasoning with ontology. The logical foundation will enable the promotion of consistent management of fundamental information flow to maintain principles linked to proactive engineering [Demoly et al., 2011b] by PDM systems and proactive definition by CAD tools within products development [Demoly et al., 2011a]. The objective can be reached by following several steps. The overall research vision is presented in Figure 2.

The first step consists of qualitatively model and describe the spatiotemporal evolution of a product in the context of AOD so that product architects’ intents can be better interpreted by designers. The description is required in the three dimensions (i.e. spatial, temporal and spatiotemporal) in order to build the novel first-order logic mereotopological theory. Region-based theories formally describe how the product is perceived in the “real world”. Indeed, this research work extends existing spatial mereotopology-based theories in the temporal and spatiotemporal dimensions in order to describe objects evolution over time and space. Moreover relationships between mechanical parts of the product can be described from an “engineering point of view” (and not from the common sense).

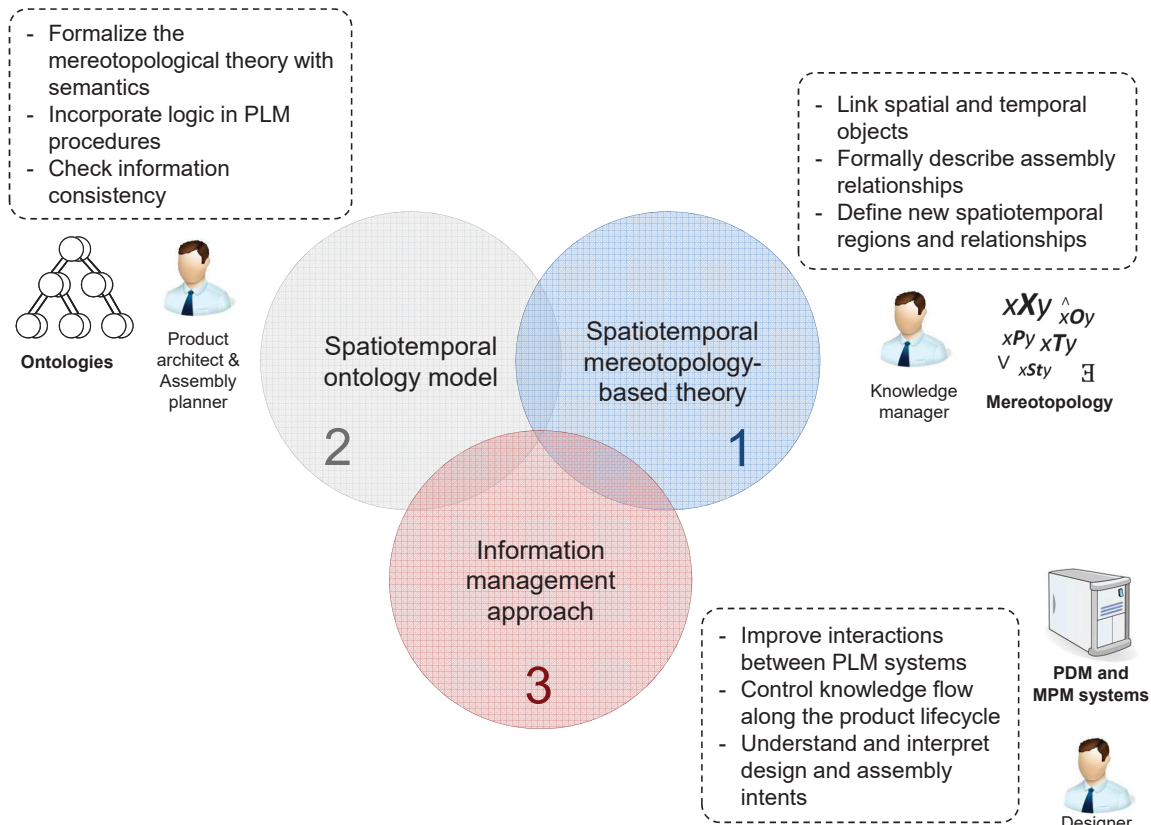


Figure 2: Proposed vision of the research

Therefore, the descriptions are more rigorous and the perspectives are more structured [Salustri, 2002].

The second step is to build a spatiotemporal ontology on product-process evolution. The ontology is implemented in Protégé Editor and uses **OWL (Web Ontology Language)**, DL and SWRL, in order to set up complex rules to reason and check the information consistency, as well as establish clear relationships between assembly components and form features [Kim et al., 2006]. The ontology explicitly represents the relationships between mechanical parts during the assembly process [Kim et al., 2006]. As such, the product and its evolution can be controlled by the ontology. Logical definition of the product will be included in the three dimensions. As logic does not require quantitative data, this description is totally adapted at the early design stages [Salustri, 2002]. Moreover, the previous theory will be formalized and will be machine-interpretable by data management systems for product (i.e. PDM) and process (i.e. MPM) and by design tools (i.e. CAD applications) [Demoly et al., 2012c].

The third step is to propose a novel approach to efficiently manage product-process information through PLM systems. The top-down design approach is based on a hub, which orchestrates information flows with the support of the previous developed ontology.

---

This approach will enable a better interaction between information systems (i.e. PDM and MPM). As such, the product evolution will be better clarified from the early design stages. Therefore, the designer's understanding on the AOD process will be facilitated and improved. A better design support will promote product architects' and designers' awareness and will increase the products quality [Zeng and Gu, 1999b].

## **1.4 Summary**

This section has introduced the industrial stakes and the scientific context, on which the research works are based. A lack of approach to manage products (evolving over time during their development) has been highlighted. Then, problem statement and research aims have been expressed so as to guide step by step our research development.

The next chapter will present the state of the art on scientific contributions in the domains of mereotopology-based theories, spatiotemporal ontologies and product lifecycle management-based approaches.



# Chapter 2

## State of the art in the domain of mereotopological theories, ontology and product lifecycle management

### Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>13</b>
<b>2.2</b>	<b>Mereotopology-based theories</b>	<b>13</b>
2.2.1	Definition	13
2.2.1.1	Mereology	13
2.2.1.2	Mereotopology versus mereogeometry	14
2.2.1.3	Four-dimensionalism	15
2.2.2	Background in Mereotopology	16
2.2.2.1	Common core of mereotopology	16
2.2.2.2	Description and classification of the mereotopology-based theories	17
2.2.3	Main approaches in product design	22
2.2.3.1	Salustri's approach	22
2.2.3.2	Approach from TU Delft	22
2.2.3.3	Kim's approach	24
2.2.3.4	Demoly's approach	24
2.2.4	Remaining challenges and emerging needs	26
2.2.4.1	Basic problems in mereotopology	26
2.2.4.2	Assembly-Oriented Design	27
2.2.4.3	Related issues and needs in engineering design and CAD knowledge representation	29
2.2.5	Summary and position	29



<b>2.3</b>	<b>Ontology models</b>	<b>31</b>
2.3.1	Ontology definition and goal	31
2.3.2	Product design rationale, intents and formalisms	32
2.3.3	Spatiotemporal ontologies	33
2.3.4	Main ontology models in product design	35
2.3.4.1	FBS model	35
2.3.4.2	PRONOIA approach	36
2.3.4.3	CPM and OAM models	36
2.3.4.4	AsD approach	39
2.3.5	Summary and position	40
<b>2.4</b>	<b>Product Lifecycle Management</b>	<b>41</b>
2.4.1	PLM definition and goal	41
2.4.2	Engineering change management	42
2.4.2.1	Changes	42
2.4.2.2	Definition and needs of ECM	44
2.4.3	Main engineering approaches to maintain consistency in product design and BIM	45
2.4.3.1	Demoly's et al. approach	45
2.4.3.2	Louhichi's and Rivest's approach	45
2.4.3.3	Chen's and Luo's approach	45
2.4.4	Heterogeneity management in other domains	46
2.4.5	Summary and position	48
<b>2.5</b>	<b>Summary: need of theory, model and approach</b>	<b>48</b>

---

## 2.1 Introduction

This chapter presents a comprehensive state of the art survey so as to clarify the different concepts presented in chapters 3, 4 and 5. The first section introduces mereotopology-based theories and classifies them. As such, theories are compared according to their domains of applications and their used primitives. The second section describes the notion of ontology and the most known models in design domain. As such, Figure 3 illustrates the decomposition structure of research areas from formal ontology to mereotopology. Gray rectangles and bold lines represent the main entities described in this chapter. Then the third paragraph introduces the different information systems, such as PLM, PDM and MPM. Moreover a review on engineering change management is provided. At the end, a conclusion presents the research needs and the position that we will adopt in the next chapters.

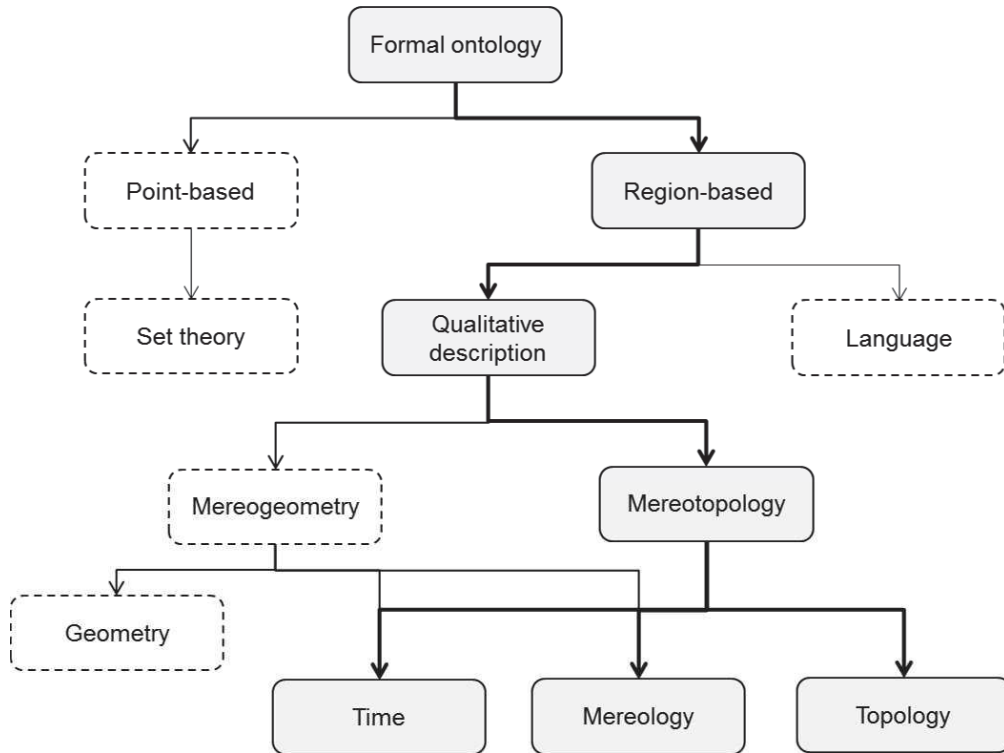


Figure 3: Tree of investigated research fields [Gruhier et al., 2014a]

## 2.2 Mereotopology-based theories

### 2.2.1 Definition

#### 2.2.1.1 Mereology

Lesniewski [Lesniewski, 1929b] was the first to work at the construction of three logical systems, called protothetic (i.e. combines the next two notions), ontology (i.e. *is\_a*) and mereology (i.e. *is\_part\_of*). Protothetic introduces for instance the logic of quantifiers such as “For all” and “For some”. Based on his previous work, Lesniewski [Lesniewski,

1929a] has introduced and described the mereology theory including theorems and axioms, in order to develop the parthood relation in a formal manner. Mereology comes from the greek and means “Part” and “Study, discussion, science”. A particular aspect is the use of *part of* primitive [Duntsch et al., 2001], therefore representing part-whole relationship. Salustri and Lockledge [Salustri and Lockledge, 1999] proposed two different thoughts in mereology. The first, which represents the real world, introduced a transitive *part of* predicate. The second one was suggested by [Artale et al., 1996] and represented the cognitive structure.

Although this theory has been recognized as a foundation for first-order logic description, several researchers have highlighted limits and basic problems, especially in application domains [Varzi, 1998] and [Salustri, 2002]. As an automotive example (cf. Figure 4), the speedometer *is part of* the dashboard, the dashboard *is part of* the car and the car *is part of* a fleet. This seems to be logical sentences, but in no case the speedometer is part of a fleet. The problem encountered here is that parthood relation is transitive (i.e. parthood means only one thing at once) in the first two statements and intransitive (i.e. parthood means something else) in the last one [Salustri and Lockledge, 1999]. To overcome this paradox, the notion of topology has been incorporated and considered together with mereology so as to initiate mereotopology.

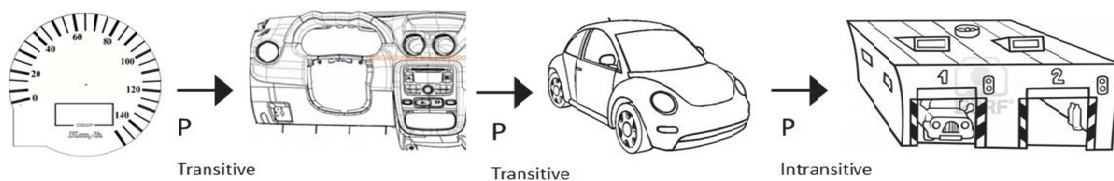


Figure 4: Representation of the “transitive problem”

### 2.2.1.2 Mereotopology versus mereogeometry

Mereotopology is a critical theory for ontological analysis [Varzi, 1998]. This theory enables the qualitative formalization of two fundamental predicates: parthood (i.e. one entity *is part of* another) and connection (i.e. an entity *is connected to* another). The current challenge of the mereotopology-based theory in engineering design is to consider the product as it is perceived in the real world [Demoly et al., 2012b].

Unlike to mereotopology, which extends mereology with topological notions, mereogeometry extends mereology with geometric concepts [Van Harmelen et al., 2007]. Moreover mereogeometry is used to describe geographical spaces [Pratt and Schoop, 2000] or to represent qualitative movement of physical bodies [Borgo et al., 1996] [Muller, 1998] [Bennett et al., 2000]. The best-known theory is called “Geometry of solids” [Tarski, 1956]. Among past research works in mereogeometry, Borgo and Masolo [Borgo and Masolo, 2008] have

Table 1: Comparison between endurantism- and perdurantism-based approaches [Al-Debei et al., 2012]

Endurantism	Perdurantism
Objects have only spatial dimensions	Objects have spatial and temporal dimensions
Objects are wholly present at any point of time during their lifetime	At any given time a 4D object is only partially present
Objects are viewed from the present	Objects from the past, present and future all exist
Objects do not have temporal parts	Objects extend in time as well as space and have temporal parts as well as spatial parts
Different objects may coincide at a point in time	When two objects have the same spatio-temporal extent, they are the same thing
Time and space are treated separately	Time and space are unified
Understand change in terms of things	Understand things in terms of change

proposed a comparative method in order to bring out a full mereogeometry. The strongest mereogeometrical theory can capture the full system of Euclidean geometry by defining points to be collections of concentric spheres.

These two region-based theories are promising mathematical descriptions, which belong to the branch of logic, first-order logic one. They are both applied in many domains such as in the semantics of natural language and in QSR. However, Hahmann [Hahmann, 2012] states that mereogeometry enables less qualitative description and is more restricted to describe classical geometries (i.e. because of the more fine-grained relationships).

### 2.2.1.3 Four-dimensionalism

Conceptual modeling enables the description of the world so as to improve the understanding of objects surrounding us. Two main metaphysical beliefs, describing modeling paradigm, are here confronted, i.e. endurantism and perdurantism. The different paradigms depend on the way we actually perceive the real world.

Indeed, endurantism considers objects as three dimensional entities that exists wholly at any given point of their life [Hales and Johnson, 2003]. On the contrary, perdurantism approach (also known as four-dimensionalism) considers that objects have four dimensions, i.e. three spatial and one temporal dimensions. The temporal part, which composed the object, is called *timeslice* [Welty and Fikes, 2006]. At each *timeslice*, the object is partially represented and a *timeslice* is only valid at a specific instant or interval [Harbelot et al., 2013]. As such, a full description of an object is the collection of all the *timeslices*. Perdurantism also fits with current scientific understanding of the world [Hales and Johnson, 2003]. Both approaches are compared in Table 1.

Besides, four-dimensionalism is defined as the description of spatial entities by including the temporal aspect. The **ISO 15926** (**I**nternational **O**rganization for **S**tandardization) [Standard ISO 15926, 2003] - used in oil and gas industry - presents time as the fourth dimension: three spatial dimensions (x,y,z) and a temporal one (t). This standard aims

at storing and exchanging lifecycle information with a strong emphasis on the representation of temporal changes [Morbach, 2009]. Efforts have been made towards temporal logic and the related primitives including motion or not to interpret paths of objects or events [Demoly et al., 2012b]. Sider [Sider, 2001] defends four-dimensionalism, as the picture of persistence over time. According to the same author, a car assembly line is like a (time, space) diagram: the horizontal axis represents the evolution of the car according to time (at the beginning it contains just the body, then the tyres and the doors are added) and the vertical axis represents the space (e.g. when the tyres are added, the car is taller, etc.). The temporal change can be seen for instance at the end of the assembly when the car is driven or is stopped.

## 2.2.2 Background in Mereotopology

This section presents the basis of the mereotopology-based theories, and proposes a list of theories in several domains. Limits and key primitives are pointed out for a potential application in engineering design, especially in product design and CAD modeling phases.

### 2.2.2.1 Common core of mereotopology

By developing such a mereotopological theory, two main notions have to be introduced, so as to describe product evolution, such as the regions (e.g. a spatial region is a portion of space occupied by some entities such as a physical part [Salustri, 2002]) and the primitives (i.e. used to describe region-to-region relationships). In other words, mereotopology describes relationships between parts with an engineering sense. Here, the engineering sense requires a more accurate and more structured prospect [Salustri, 2002] such as required in AOD.

By considering Lesniewski's work [Lesniewski, 1929b], Leonard and Goodman [Leonard and Goodman, 1940] kept on working on mereotopology and created the calculus of individuals, which is become the basis for the study of formal part-whole relations. Clarke [Clarke, 1981] based his theory on the classical mereology and also on Whitehead's results [Whitehead, 1929] and presented an axiomatic system for a calculus of individuals. In the last fifteen years, mereotopology has met success especially in the fields of qualitative spatial reasoning.

Entities, that exist in other spaces besides the usual one, can actually be represented with region-based theories, such as developed by Eschenbach et al. [Eschenbach et al., 1994] and Randell and Cohn [Randell and Cohn, 1989]. A region is defined as a portion of space occupied by some entities or material (for a physical part) or something else (e.g. a hole). The overarching goal of any mereotopological theory is to describe the nature of regions, the entities which occupy them and the interrelations between regions [Salustri, 2002]. The numerous identified theories do not use the same primitives, but a common core can be stressed. Mereotopological operators (cf. Table 2) have been used along the paper. According to Varzi [Varzi, 1998] and Salustri [Salustri, 2002], the parthood primitive  $\mathbf{P}$  is reflexive ( $\forall x x\mathbf{P}x$ ), transitive ( $\forall xyz x\mathbf{P}y \wedge y\mathbf{P}z \rightarrow x\mathbf{P}z$ ) and

Table 2: Fundamental mereotopological operators [Smith, 1996]

Symbol	Name
$\wedge$	Logical conjunction
$\vee$	Logical Disjonction
$:=$	Definition
$\rightarrow$	Logical implication
$\exists$	Existential quantifier
$\neg$	Logical negation
$\forall$	Universal quantifier
$\emptyset$	Empty region
$A$	Overall assembly
$\phi$	Condition
$\sigma$	Sum (fusion or join)
$\equiv$	Equivalence
$\iota$	Definite descriptor

anti-symmetric ( $\forall xy \ xPy \wedge yPx \rightarrow x = y$ ). According to Muller [Muller, 1998], the connection primitive is also reflexive and transitive, but it is symmetric ( $\forall xy \ xCy \rightarrow yCx$ ).

### 2.2.2.2 Description and classification of the mereotopology-based theories

Smith [Smith, 1996] has adopted as mereological primitive the relation of parthood and as topological primitive, the relation “*is an interior part of*”, but also various other primitives such as seen in Figure 5. He has addressed a great importance on how to represent boundaries between spatial regions. The latter have no Interior Parts (**IP**) and they do not exist independently of the entities they bound.

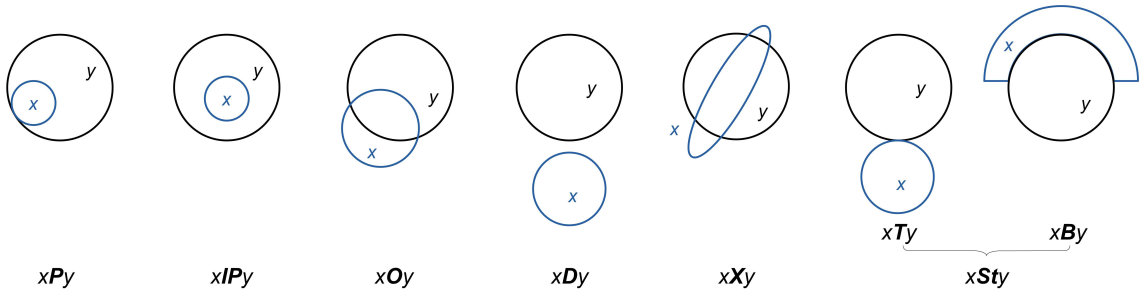


Figure 5: Representation of the Smith's mereotopological primitives [Smith, 1996]

They can be divided into exterior and interior (the exterior boundaries of  $x$  being boundaries which separate  $x$  from the remainder of the universe). Based on this, Smith defines additional primitives such as  $x$  crosses  $y$  ( $xXy = \neg xPy \cap \neg xDy$ ),  $x$  straddles  $y$



Table 3: RCC8 mereotopological primitives description

Name	Mereotopological description
$x$ DisConnected to $y$	$DC(x, y) = \neg C(x, y)$
$x$ is Part of $y$	$P(x, y) = \forall z (C(z, x) \rightarrow C(z, y))$
$x$ Overlaps $y$	$O(x, y) = \exists z (P(z, x) \wedge P(z, y))$
$x$ is Proper Part of $y$	$PP(x, y) = P(x, y) \wedge \neg P(y, x)$
$x$ Discretes $y$	$D(x, y) = \neg O(x, y)$
$x$ is Externally Connected to $y$	$EC(x, y) = C(x, y) \wedge \neg O(x, y)$
$x$ is Tangential Part of $y$	$TP(x, y) = P(x, y) \wedge \exists z (EC(z, x) \wedge EC(z, y))$
$x$ is Non Tangential Part of $y$	$NTP(x, y) = P(x, y) \wedge \neg \exists z (EC(z, x) \wedge EC(z, y))$

( $xSty = \forall z (xIPz \rightarrow zXy)$ ,  $x$  is the boundary of  $y$  ( $xBy = \forall z (zPx \rightarrow zSty)$ ) and  $x$  is self-bounding ( $xBy = \neg \exists t (tIPx)$ ).

Randell, Cui and Cohn [Randell et al., 1992] have developed **RCC** (**R**egion **C**onnection **C**alculus). Here, the first primitive is the connection and parthood is defined from it. Eight relations are then defined (cf. Table 3) and represented (cf. Figure 6). Here, **TPP** means *Tangential Proper Part*, **TPPI** *Tangential Proper Part Inverse* and **NTPP** as well as **NTPPI** are the respective negations.

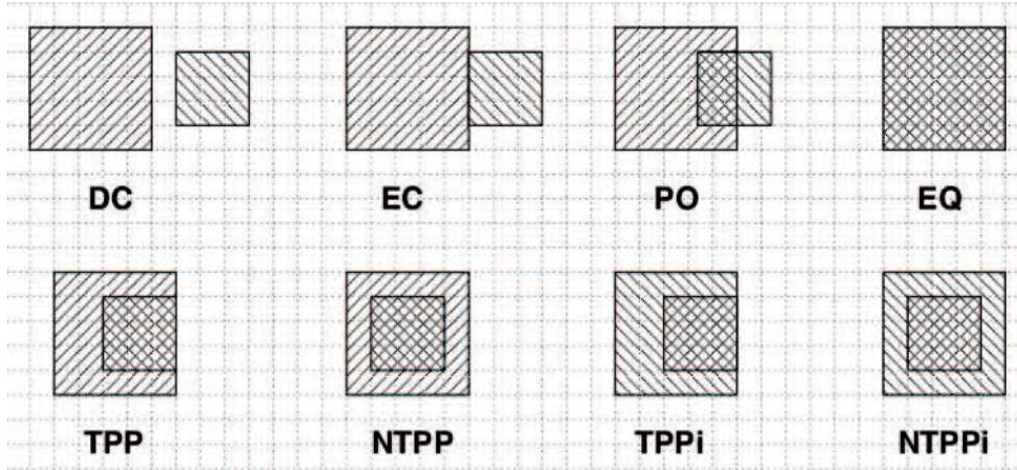


Figure 6: The eight RCC8 relations [Randell et al., 2012]

In addition, Asher and Vieu [Asher and Vieu, 1995] define the parthood based on the connection primitive. Unlike Clarke's theory [Clarke, 1981], Asher and Vieu's theory is a first order one: the explicit fusion operator is eliminated and the concept of weak contact (i.e. two objects or regions touch without being fully connected) is added [Hahmann, 2008]. For instance weak contact can be found between a tyre and a car, since space exists between them. A way to eliminate weak contact is to refine the granularity of the

space.

According to Varzi [Varzi, 1996], the notion of connexity cannot be defined by mereology, and so it becomes mandatory to introduce the connection primitive. The same author states that several kinds of mereology exist such as: the ground mereology, extensional mereology (i.e. with supplementation axiom), closed mereology (i.e. with the sum, product and difference primitives) and classical mereology (i.e. with the fusion axiom).

To be able to reason efficiently about spatial qualitative models, Muller [Muller, 1998] has used and implemented composition tables. As such, several motion classes have been introduced such as **LEAVE**, **REACH**, **HIT**, **CROSS**, **INTERNAL**, and **EXTERNAL**, which are represented in Figure 7.

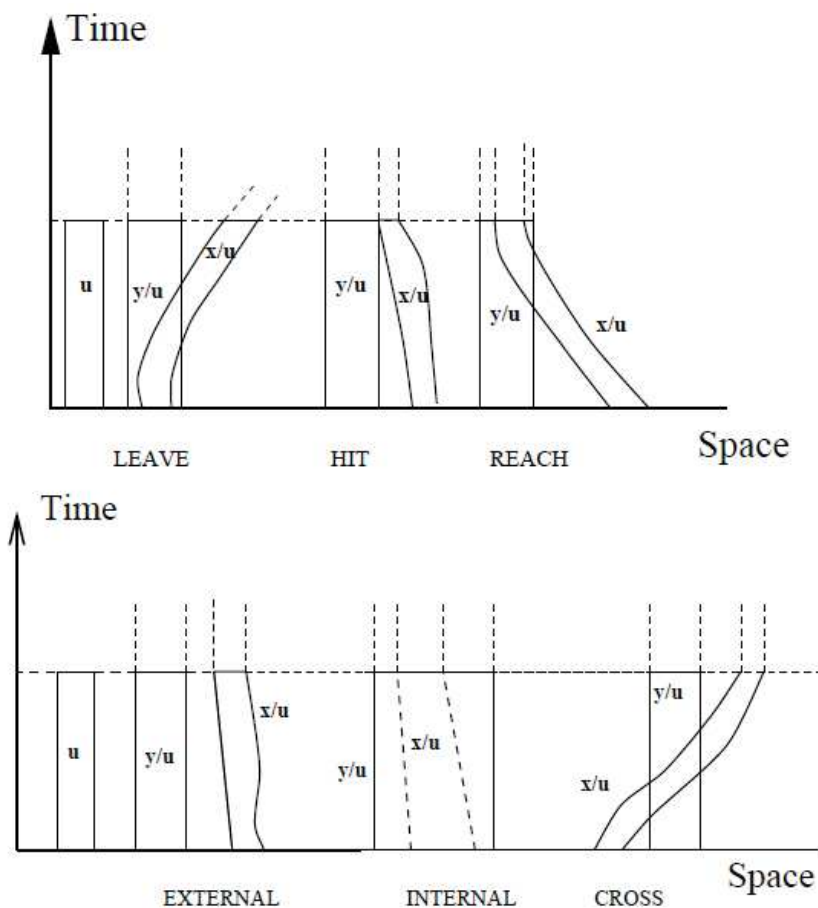


Figure 7: The six motions classes of Muller [Muller, 1998]

Moreover Clarke [Clarke, 1985] formalizes the notion of point as a limit of regions interlocking. By using the fusion operator ( $\oplus$ ), the author defines intersection, complement, interior and closed regions. Thus, two kinds of connection exist: connection with



external contact and connection with covering. In addition, Asher and Vieu [Asher and Vieu, 1995] have demonstrated that Clarke’s assumptions were wrong.

Recently, Stell and West [Stell and West, 2004] have developed a novel theory based on boolean connection algebra, in which two additional structures are proposed: a topological closure operator ( $\diamond$ ) and a pair of operators forming a Galois connection (*GAL*), to define temporal relations between spatiotemporal entities. Moreover the concept of historical closure is also described to define temporal part, historical part and historical connection. The historical closure is a spatiotemporal region and consists of all space and time during the existence of the considered region. Figure 8 presents the historical closure (i.e. solid black region) composed of two disconnected parts. A significant simplification has been achieved compared to the Muller’s approach. As a result, three distinct primitives are introduced: spatiotemporal part, temporal part and historical part.

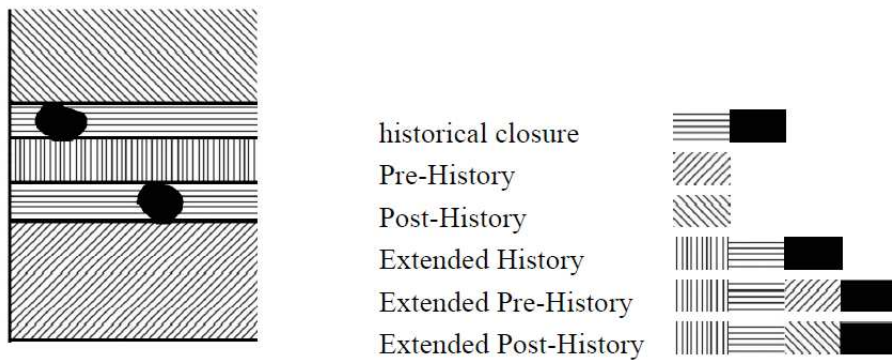


Figure 8: Visualization of Stell and West’s operators [Stell and West, 2004]

A proposed classification of the existing theories in product design is presented in Table 4 according to authors, their first spatial primitives on which theories are based, developed primitives (which are described according to the first one) in spatial, temporal and even spatiotemporal dimensions and their application domain [Gruhier et al., 2014b]. As such, the classification of primitives has been established which highlights missing parts to fully cover product-process design domain.

Besides, it seems that most of the listed theories does not cover entirely spatial and temporal dimensions, but rather provides an accurate description of spatial dimension for a specific domain. A novel dimension in mereotopological theory is required insofar as objects evolutions have to be described over space and time. Moreover temporal description is not applied to product design information, which means that evolution in product definition and lifecycle sequence knowledge is not considered. This table represents which primitives are used by all authors in the mechanical domain. It seems that primitives such as *Part of*, *InteriorPart of*, *Overlap*, *Tangent* or their equivalents are used by every author in the area.

Table 4: Classification table of relevant region-based theories [Gruhier et al., 2014b]

Author	Dimension			Application domain
	Spatial		Temporal	
	First primitive	Developed primitive	Spatiotemporal	
[Allen, 1983]			<; =	Artificial intelligence
[Asher and Vieu, 1995]	<i>C</i>	<i>P; PP; O; EC; TP; NTP</i>		Artificial intelligence
[Bennett et al., 2000]	<i>P; S</i>	<i>PP; D; O; PO; B; COB; EQD; MID; NEARER</i>		Artificial intelligence
[Borgo et al., 1996]	<i>P; SR; CG</i>			Knowledge representation and reasoning
[Clarke, 1981]	<i>C</i>		< $\subseteq t$	Mathematics
[De Laguna, 1922]	<i>C</i>		,	Philosophy
[Demoly et al., 2012b]	<i>P</i>	<i>O; IP; D; Point; X; St; B; T; IB</i>	< <i>Ot; <math>\subseteq t</math>; *</i>	Mechanical assembly
[Duntsch et al., 2001]	<i>C</i>	<i>P; PP; O; PO; TPP; NTPP; EC; DC</i>		Computer Science
[Eschenbach, 1999]	<i>P; DC</i>	<i>PP; O; PO; D; C; EC; TPP; NTPP; component</i>		Cognitive science
[Galton, 1997]	<i>C</i>	<i>DC; P; O; D; C; EC; PO; EQ; PP; TPP; NTPP</i>	<; =	Knowledge representation and reasoning
[Goodman, 1951]	<i>D</i>	<i>P; O</i>		Mathematics
[Kim et al., 2008]	<i>P; IP</i>	<i>O; D; Point; X; St; B; T</i>		Mechanical joint
[Lemon and Pratt, 1998]	<i>C</i>			Visual languages and computing
[Lesniewski, 1929a]	<i>I</i>			Mathematics
[Muller, 1998]	<i>C</i>	<i>P; PP; O; PO; EC; TP; NTP; TPP; NTPP; TPPI; NTPPI</i>	< <i>TS; <math>\subseteq t</math>; *</i>	Knowledge representation and reasoning
[Randell and Cohn, 1989]	<i>C</i>	<i>DC; P; O; PP; D; EC; TP; NTP; TPP; NTPP; TPI; NTPI; PO</i>	<i>B</i>	Knowledge representation and reasoning
[Salustri, 2002]	<i>P; C</i>	<i>PP; O; EC; TPP; NTPP; SC; E</i>		Mechanical engineering
[Smith, 1996]	<i>P; IP</i>	<i>O; D; Point; X; St; B</i>		Mechanical engineering
[Stell and West, 2004]	<i>C</i>	<i>P; PP; O; EC; NTPP</i>	< <i>Pt; GAL; Hc</i>	Information system
[Tarski, 1956]	<i>P; S</i>			Mathematics
[Varzi, 1996]	<i>P; C</i>	<i>PP; O; U; OX; UX; PO; PU; EC; TP; IP; E; IE; TE; SU; SX; PS</i>		Philosophy
[Whitehead, 1929]	<i>R; C</i>			Mathematics and philosophy

$<$  (Precedence),  $\subseteq t$  (temporal inclusion),  $*$  (temporal connection), *B* (Wholly before), *C* (Connected), *D* (Discrete), *E* (Enclosure), *EC* (Externally Connected), *IB* (Internal Boundary), *IP* (InteriorPart), *NTPP* (Non Tangential Proper Part), *O* (Overlap), *Ot* (temporal overlap), *P* (Part of), *Point*, *X* (Cross), *PP* (Proper Part), *SC* (Self-Connection), *St* (Straddle), *T* (Tangent), *TPP* (Tangential Proper Part).

By focusing on mechanical engineering application, few research efforts have been done and at different abstraction levels (i.e. functional, kinematic and technological aspects). More precisely, Randell and Cohn [Randell and Cohn, 1989] research works have enabled the representation of relationships between fluid and solid objects at different states. Besides, Demoly et al. have described product relationships by mainly focusing on kinematic pairs. Kim et al. have proposed a region-based theory to differentiate assembly joints obtained by welding or adhesive bonding. They both used Smith's mereotopological primitives [Smith, 1996]. Finally, Salustri has also introduced similar primitives, which have the same properties but not the same names (e.g. *InteriorPart* **IP** is equivalent to *Non Tangential Proper Part* **NTPP** or *Tangential Proper Part* **TPP**), so as to cover the product modeling issue. Moreover he did not use the *Crosse* **X** primitive, which seems to be relevant in the above mentioned context and which is intensively used during assembly sequence planning phase.

### 2.2.3 Main approaches in product design

Here, some approaches, which are close to our research domain of interest, will be described in details to highlight the key information concerning mereotopological theories.

#### 2.2.3.1 Salustri's approach

Salustri [Salustri, 2002] has developed a **Design Mereotopology (DMT)**, in order to improve the understanding of product modeling knowledge and have better computer-based aids. DMT is based upon the **Closed Region Calculus (CRC)** [Eschenbach, 1999] and the theory presented in Smith [Smith, 1996]. It is a formal theory of reasoning, which expresses an "engineering sense" (as distinct from common sense). Some domains of application are enumerated below:

- *Computer-based design tools* - to support designers in building product models with the development of **KBS (Knowledge-Based System)** using description logics;
- *Geometric and configuration modeling* - to be used especially in the upstream stages of design process;
- *Material knowledge-bases* - to differentiate materials, which are represented with regions. As such, a void in the graph indicates a combination of characteristics for which no material exists;
- *Function modeling* - to model product function represented on an axis in space. As such, product is described as a region in the function-space graph;
- *System modeling* - to represent physical objects that provide a set of functions.

#### 2.2.3.2 Approach from TU Delft

TU Delft, with the support of Prof. Horváth, has worked on two different approaches to describe the product: Nucleus-based conceptualization and Mereotopological theory.

The first approach uses nucleus so as to model and simulate virtual human artefact systems. Van der Vegte et al. [Van der Vegte et al., 2008] have chosen to use nucleus to model entity composing of one or two regions. Nucleus has been identified to describe large deformations of flexible objects. As such, an object is discretized into particles, which are interrelated with relationships [Horvath and van Der Vegte, 2003]. In nucleus-based modeling, three groups of reflexive relations have been defined: existence, reference and substance. Moreover, eight groups of binary relations are also defined: connectivity, positioning, morphological, kinematical, deformation, kinetic, physical phenomenon and physical field.

As such, the User (U) can be described, as well as the Product (P) and the Environment (E) (cf. Figure 9). The nucleus is the lowest level modeling entity that can be used equally well in representing the three actors [Van der Vegte and Horvath, 2003].

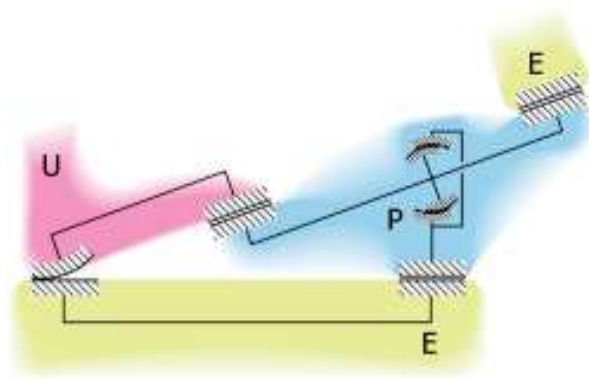


Figure 9: Nucleus-based U-P-E model [Van der Vegte and Horvath, 2003]

Horváth and Van der Vegte [Horvath and van Der Vegte, 2003] consider the nucleus theory as the foundation and the next generation of new product modeling methodology. The proposed theory offers a relation-oriented modeling in the early design stages (when the geometry is not yet defined). As such, a set of relations, specified in terms of attributes, parameters and descriptors, is provided to the designer. In addition, the nucleus theory gives the opportunity to represent the history of physical effects together with the changes in a product.

The second approach is based on mereo-operandi theory. Its essence is a combination of [Pourtalebi et al., 2014]:

- **CPS** (Cyber-Physical Systems) knowledge;
- Mereotopological knowledge (e.g. to describe structural and operational architecture of CPS);
- Modus operandi (refers to operation, functionality and behavior) knowledge.

### 2.2.3.3 Kim's approach

Kim et al. [Kim et al., 2008] have used mereotopological theories to describe assembly joints. The main purpose is to differentiate them with ontology rules in collaborative and intelligent product design. In assembly process, joints are a key aspect. In fact, assembly models are often ambiguous when model sharing takes place. The physical implications can differ according to the joining processes, even if joints have similar geometries and topologies. Based on Smith's spatial mereotopological primitives [Smith, 1996], Kim et al. [Kim et al., 2004] have managed to develop an assembly design formalism, with a region-based theory for parts. As such, assembly relations are defined and a meta-model called **ARM** (**A**ssembly **R**elation **M**odel) is generated in order to explicitly represent the relations in an XML format. Assembly joints (such as fusion welding) are then formally defined. Below is the mereotopological definition of fusion welding:

$$x\mathbf{J}_{fw}y := \exists w(w\mathbf{O}x \wedge w\mathbf{O}y) \wedge \exists w(w\mathbf{St}j \vee w\mathbf{X}j) \quad (2.1)$$

This definition indicates that if there is an entity,  $w$ , associated to joining, we observe that the entity  $w$  (i.e., a weld) overlaps  $x$  and  $y$ , which are fusion welded. In addition, the  $w$  should either straddle or cross to  $j$ , the mating boundary as illustrated in Figure 10. Here  $j$  is a virtual geometric entity.

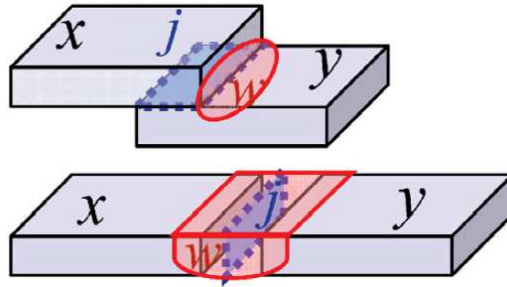


Figure 10: Exemple of fusion welding [Kim et al., 2008]

$$j := \sigma z(\phi z) \rightarrow \forall z(\phi z \rightarrow z\mathbf{B}x \wedge z\mathbf{B}y) \quad (2.2)$$

Figure 11 sums up different joining methods and their related mereotopological definitions, their illustrations and their descriptions. A total of eight joints have been described. Here rivet and metal stitching descriptions are quite ambiguous, as the same definition is obtained with mereotopology.

### 2.2.3.4 Demoly's approach

Demoly et al. [Demoly et al., 2012d] describe the product relationships in the context of AOD by using the part-whole theory supported by mereology and its extension mereotopology. Based on Smith's spatial primitive [Smith, 1996], Demoly et al. [Demoly et al., 2012b] have extended the theory in the temporal and spatiotemporal dimensions

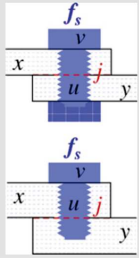
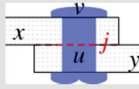
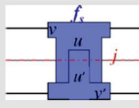
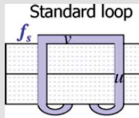
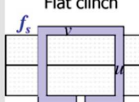
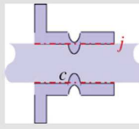
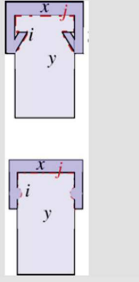
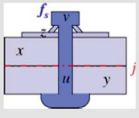
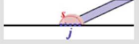
Joining method	Mereotopological definition	Illustrative example	Description
Mechanical fastening by threaded fasteners	$\lambda j_e v: = \exists u, v \{ (uXx \wedge uXy) \wedge (vTx \vee vTy) \wedge (uPf_s \wedge vPf_s) \} \wedge f_s Xj$		Mechanically fastened joints. The most common method of mechanical fastening is by using bolts, nuts, screws, pins, and a variety of other fasteners $u$ = threaded body $v$ = head of fastener and/or nut
Riveting or mechanical fastening by rivets	$\lambda j_e v: = \exists u, v \{ (uXx \wedge uXy) \wedge (vTx \wedge vTy) \wedge (uPf_s \wedge vPf_s) \} \wedge f_s Xj$		Installing a rivet consists of placing the rivet in the hole and deforming the end of its shank by upsetting or heading $u$ = body of rivet $v$ = head and upset tail of rivet
Compression	$\exists u, v \{ (uXx \wedge uXy) \wedge (vTx \wedge vTy) \wedge (uPf_s \wedge vPf_s) \} \wedge f_s Xj \wedge \exists u', v' \{ (u'Xx \wedge u'Xy) \wedge (v'Tx \wedge v'Ty) \wedge (u'Pf_{s'} \wedge v'Pf_{s'}) \} \wedge f_{s'} Xj \wedge \exists u, u' (uTu')$		$u$ = body of fastener $v$ = head of fastener $u'$ = body of counter fastener $v'$ = head of counter fastener
Metal stitching/stapling	$\lambda j_e v: = \exists u, v \{ (uXx \wedge uXy) \wedge (vTx \wedge vTy) \wedge (uPf_s \wedge vPf_s) \} \wedge f_s Xj$	 	Metal stitching or stapling is much like that of ordinary stapling of papers $u$ = body of staple $v$ = head of staple
Crimping	$\exists c (cPx \wedge cStx \wedge cXj \wedge cTy) \wedge xTy$		A method of joining without using a fastener. It can be done with beads or dimples, which can be produced by shrinking or swaging operations
Snap-in	$\exists i (iBx \wedge iTy) \wedge jTy \wedge xSty$		$i$ = inner surface of $x$ $y$ is inserted into a cap, $x$
Push-on fastener	$\exists u, v \{ (uXx \wedge uXy \wedge uXz) \wedge (vTx \wedge vTz) \wedge (uPf_s \wedge vPf_s) \} \wedge zTy \wedge f_s Xj$		$u$ = body of fastener $v$ = head of fastener $z$ = supporting plate
Soldering	$\exists s (jPs \wedge sStx \wedge sSty)$		$s$ = solder

Figure 11: Mereotopological representations of some assembly joints [Kim et al., 2008]

by considering Muller [Muller, 2002] and Stell and West's [Stell and West, 2004] efforts. It leads to the introduction of new primitives, such as represented in Figure 12. These primitives enable the definition of temporal relations between spatial regions.



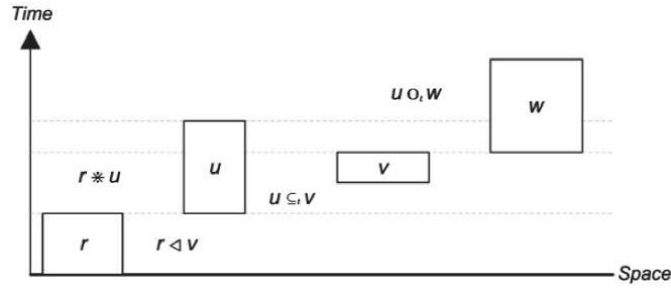


Figure 12: Representation of temporal relations for regions  $r$ ,  $u$ ,  $v$  and  $w$  [Demoly et al., 2012b]

Table 5: Mereotopological description of kinematic pairs Mereotopological description of kinematic pairs [Demoly et al., 2012b]

Kinematic pair	Mereotopological description	Comment
Rigid	$x\mathbf{K}_{\text{rig}}y := \exists x, y \{ (x\mathbf{X}y \wedge x\mathbf{T}y) \wedge (x\mathbf{O}k \wedge y\mathbf{O}k) \}$	$k$ is a point
Revolute	$x\mathbf{K}_{\text{rev}}y := \exists x, y \{ (x\mathbf{X}y \wedge x\mathbf{T}y) \wedge (x\mathbf{O}k_1 \wedge y\mathbf{O}k_1) \wedge (x\mathbf{T}k_2 \wedge y\mathbf{T}k_2) \}$	$k_1$ is a line, $k_2$ is a plane
Prismatic	$x\mathbf{K}_{\text{pri}}y := \exists x, y \{ x\mathbf{X}y \wedge (x\mathbf{O}k \wedge y\mathbf{O}k) \}$	$k$ is a line
Screw	$x\mathbf{K}_{\text{scr}}y := \exists x, y \{ (x\mathbf{X}y \wedge x\mathbf{T}y) \wedge (x\mathbf{O}k_1 \wedge y\mathbf{O}k_1) \wedge (x\mathbf{T}k_2 \wedge y\mathbf{T}k_2) \}$	$k_1$ is a line, $k_2$ is a plane
Cylindrical	$x\mathbf{K}_{\text{cyl}}y := \exists x, y \{ x\mathbf{X}y \wedge (x\mathbf{O}k \wedge y\mathbf{O}k) \}$	$k$ is a line
Spherical	$x\mathbf{K}_{\text{sph}}y := \exists x, y \{ (x\mathbf{S}t y \wedge x\mathbf{I}B y) \wedge (x\mathbf{O}k \wedge y\mathbf{O}k) \}$	$k$ is a point
Planar	$x\mathbf{K}_{\text{pla}}y := \exists x, y \{ x\mathbf{T}y \wedge (x\mathbf{T}k \wedge y\mathbf{T}k) \}$	$k$ is a plane
Point-contact	$x\mathbf{K}_{\text{pt}}y := \exists x, y \{ x\mathbf{T}y \wedge (x\mathbf{T}k \wedge y\mathbf{T}k) \}$	$k$ is a point
Line-contact	$x\mathbf{K}_{\text{lin}}y := \exists x, y \{ x\mathbf{T}y \wedge (x\mathbf{T}k \wedge y\mathbf{T}k) \}$	$k$ is a line
Curve-contact	$x\mathbf{K}_{\text{cur}}y := \exists x, y \{ x\mathbf{B}y \wedge (x\mathbf{B}k \wedge y\mathbf{B}k) \}$	$k$ is a curve

These descriptions are part of the PRONOIA approach. PRONOIA 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. As such, information like kinematic pairs have been converted into mereotopological equations (cf. Table 5).

## 2.2.4 Remaining challenges and emerging needs

This section describes the remaining challenges in mereotopology and related emerging needs in engineering design, especially in product conceptual design and CAD modeling stages.

### 2.2.4.1 Basic problems in mereotopology

The above descriptions and discussions of mereotopology-based theories have provided interesting foundations for spatial dimension. But currently, it seems that these theories encounter some recurring problems due to their extension towards spatiotemporal

mereotopology. For instance, Salustri [Salustri, 2002] has raised an open issue about the best way to consider the boundaries (i.e. real entities or just cognitive artefact of human perception). According to Varzi [Varzi, 1998], some principles of which the extensionality, the fusion and the external contact, are the source of some controversy. Casati and Varzi [Casati and Varzi, 1999] have worked on the existence of hole among parts.

Demoly et al. [Demoly et al., 2012b] have stressed issues to describe relationships among product parts and sub-assemblies and also to refine CAD models with a temporal perspective of the product through its lifecycle. Unlike spatial mereotopology, few efforts have been made regarding spatiotemporal mereotopology. In addition, region-based theories have been also used in **Geographic Information System (GIS)** to describe the brambles propagation over time [Del Mondo et al., 2010] and also the evolution of the flock location. Actually GIS tries to represent and understand different spatiotemporal evolutions such as the move (i.e. when an object moves from one location to another one), the geometric change (for instance the deformation of an object under pressure and the transformation or the change of geometry). Moreover, the creation, loss, fusion and fission of geographic elements are also considered as spatiotemporal relationships. More specifically in the environment, land management and geo-marketing fields, **MADS (Modeling Application Data with Spatiotemporal features)** model [Parent et al., 2006] has been introduced as a conceptual design approach for spatiotemporal database. It describes, denotes, relates and manipulates objects, which represent the real-world entities.

This latter is quite important, as the purpose of this research work is to find the most appropriate assembly representation in product conceptual and CAD definition stages by developing a dedicated mereotopological foundation. For example in mechanical engineering, product design and assembly process can be described by spatial regions (e.g. product parts and subassemblies), temporal regions (e.g. assembly operations) and spatiotemporal regions (e.g. swept volume during a part deformation or a part move). As a consequence product design engineering needs to be more rationalized and formalized [Zeng and Gu, 1999b]. If information and knowledge are defined in a way that product-process information and knowledge are machine-interpretable, then computers can easily reason on them [Kusiak and Salustri, 2007].

#### 2.2.4.2 Assembly-Oriented Design

The concept of AOD needs to be well understood before studying how to formalize product and process knowledge. The concept of integrating **ASP (Assembly Sequence Planning)** in product design was introduced at the beginning of the previous decade in order to overcome the current limitations of **Design For Assembly (DFA)** and ASP approaches. Based on detailed product geometry, and a part-to-part oriented evaluation, DFA approaches generally lead to a redesign of products. In such a context, the issue of concurrent product design and ASP [Clarke, 1981; Zha and Du, 2002], also called AOD [Descartes, 1996], has received much attention in research work during the last decade [De Laguna, 1922; Wang et al., 2008]. Product design and assembly sequence planning phases are normally undertaken separately and sequentially, which results in missing the efficient integration



between these both phases. Here, the assembly-oriented practice of product development can be considered as a top-down approach by proactively considering the assembly related product design and its relationship issues in the early phases of the product development process. AOD gives the access to designers of knowing the ever-evolving design information. As such, manufacturing process can be executed in parallel with product design. Furthermore, with this information, manufacturing will become better equipped to handle concurrent product development and better able to adapt to in-process engineering changes [Parametric Technology Corporation, 2006].

Over the last decade, numerous data models and ontologies have been built to represent various engineering fields and product lifecycle stages. Product model generally aims at describing data and information at different abstraction levels through its lifecycle [Demoly et al., 2010]. These models also consider stakeholders viewpoints covering the project life. In the AOD context, models can be generic, functional, business, multi-view and so on [Demoly, 2010]. Some of them are presented below.

Bourjault was the first to propose functional relation through a directed graph [Bourjault, 1984]. His methodology has been reused and improved by De Fazio and Whitney [De Fazio and Whitney, 1987]. In addition, an AND/OR graph, whose purpose is to facilitate the assembly sequences generation, has been presented by Homem de Mello and Sanderson [Homem de Mello and Sanderson, 1991]. In parallel, research works have been carried out on matrix-based modeling models in order to define and analyze assembly relationships for ASP [Santocchi and Dini, 1992].

Moreover, new contributions on knowledge models and semantics in assembly formalisms have been introduced. As such, Zha and Du have built a knowledge-based system using multi-agent system and Petri Net to support assembly design and ASP by considering a start from part relational information [Zha and Du, 2002]. A knowledge-based ASP approach has also been promoted by Dong et al.. Here the assembly is modelled as a connection-semantic-based assembly tree [Dong et al., 2007].

Furthermore, Kim et al. proposed a spatial relationships-based assembly design formalism describing assembly relations, and an assembly relation model, in which relations were represented in XML format [Kim et al., 2004]. This approach was not interpretable by **Computer-Aided Design (CAD)** tools, but was only focused on the geometric aspect of the product. Later, they have presented an ontology-based representation for assembly joints in collaborative product design by using mereotopological primitives and SWRL formalism [Kim et al., 2008].

More recently, Demoly et al. have described a novel product-process data management approach by introducing the management of product relationships at various abstraction levels and in separate manner [Demoly et al., 2012d]. Four kinds of product relationships (i.e. contact, precedence, kinematic pair, and technological pair) and the related associations in PLM systems have been presented. All these relationships have been represented through a multi-view model called **MUVOA (Multiple Viewpoints Oriented Assembly)**

[Demoly et al., 2010] which has been implemented in a PLM-based application.

#### 2.2.4.3 Related issues and needs in engineering design and CAD knowledge representation

Up to now, the research in formal knowledge representation within CAD applications has been carried out only by few researchers such as Kim et al. [Kim et al., 2008], where assembly joints are defined with mereotopological primitives and ontologies, and Demoly et al. [Demoly et al., 2012b], where product relationships and more specifically kinematic pairs are described by using specific primitives and axioms, and assembly sequences. Moreover Salustri [Salustri, 2002] has proposed a design mereotopology and given the related stakes in engineering. The same author considers mereology as an interesting thing since human is only aware of his models about reality, which are based on its own perceptions, and parthood relations constitute an important part of the engineer's cognitive structures. A possible application of mereotopology can therefore be found in product modeling stage, as it succeeds to capture the characteristics of the physical entities in the actual world [Salustri, 2002] [Demoly et al., 2012b].

Mereotopology may aid in developing more effective knowledge-based systems. For instance, Kusiak [Kusiak, 2006] proposed a system for predicting product cost using historical design data. Intelligent decision-support systems are especially useful in product design engineering because of high complexity associated with the decisions and of the risks associated with making wrong decisions [Kusiak and Salustri, 2007].

Mereotopology can also aid the designer by describing where the different parts of the product are placed each other at different instants and the nature of the connections between them. Such issue will improve designer's understanding and activities and provide suitable qualitative support during product design. By looking at the different regions, the designer will be aware at the early design phase if two regions (parts of the product or tools needed to fasten it) will collide during the assembly process.

In the future the purpose is to know all the product information, such as how, when and why one item *is a part of* or *is connected to* another, in order to be able to describe the product relationships at various abstraction levels through spatiotemporal dimension. This will provide an appropriate support for product architects and designers to understand how knowledge from lifecycle sequence planning is incorporated in product design and CAD models (i.e. geometric definition of the product). With such qualitative description, a small design evolution will be verified, reasoned and captured in order to be linked with the related lifecycle stage (i.e. manufacturing, assembly, disassembly, etc.).

### 2.2.5 Summary and position

This section has introduced a state of art survey in mereotopological-based theories covering the fields of mathematics, philosophy, artificial intelligence, knowledge representation and mechanical engineering. Based on a literature study, a classification table has been

introduced where investigated theories are considered according to main spatial primitive on which other ones are developed and the related application domain. The authors, working in the context of AOD, have been extracted from the classification table and retrieved in Figure 13. This map positions our proposed theory compared to Demoly et al. [Demoly et al., 2012b], Kim et al. [Kim et al., 2008], Randell and Cohn [Randell and Cohn, 1989] and Salustri [Salustri, 2002]. It will be built for designers to work in the early design stages. It will also be adapted to mechanical engineering by integrating the “engineering sense” [Zeng and Gu, 1999b; Salustri, 2002] and extended with spatiotemporal regions and primitives.

Built on this, the need to develop such a mereotopological theory in product design and modeling has been emphasized, especially to provide a comprehensive, formal and machine-interpretable representation of product-process information and knowledge at various abstraction levels from conceptual to detailed design stages. As the purpose is to get the designer intent as early as possible in the design process, primitives such as used by Demoly et al. or Kim et al. will be a strong basis to build the theory. With mereotopology, designers will know the description of the object evolution in assembly design. The theory will also contribute to know if entities have a lack of consistency. In such a way, designers will work with a full insight and understanding of product definition and assembly.

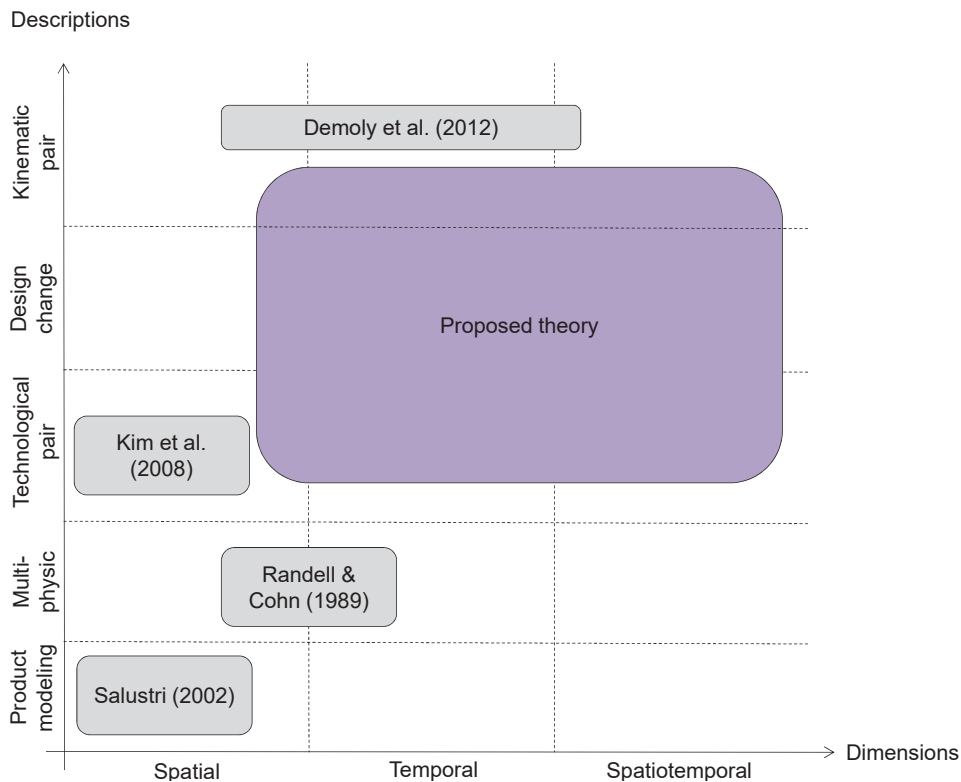


Figure 13: Map of existing mereotopological theories in the domain of assembly-oriented design

## 2.3 Ontology models

Now that mereotopological theories have been reviewed, ontologies are investigated in order to formalize the theory and be able to check the information consistency.

### 2.3.1 Ontology definition and goal

Semantic Web has been introduced in the early 2000s in the field of computer science and internet [Shadbolt et al., 2006]. It is an additional layer to current web, which makes the content of web resources accessible and usable by computers. Unlike the current web, Semantic Web works with a list of knowledge (instead of a list of data) and searches with concepts (instead of keywords). As such, web navigation will include reasoning to just get the information expected during the research. Figure 14 presents the different layers composing the Semantic Web. At the bottom, the unicode (i.e. a basic alphabet), the **URI** (i.e. **U**niform **R**esource **I**dentifiers) and the XML (i.e. a common language) layer are fully standardized. Then, RDF (i.e. triplets that connect objects together), ontology and logic are currently in progress for the standardization. At the top of the pyramid, proof (i.e. to know how things work) and trust (e.g. trust of the web information) are still in the experimental stage.

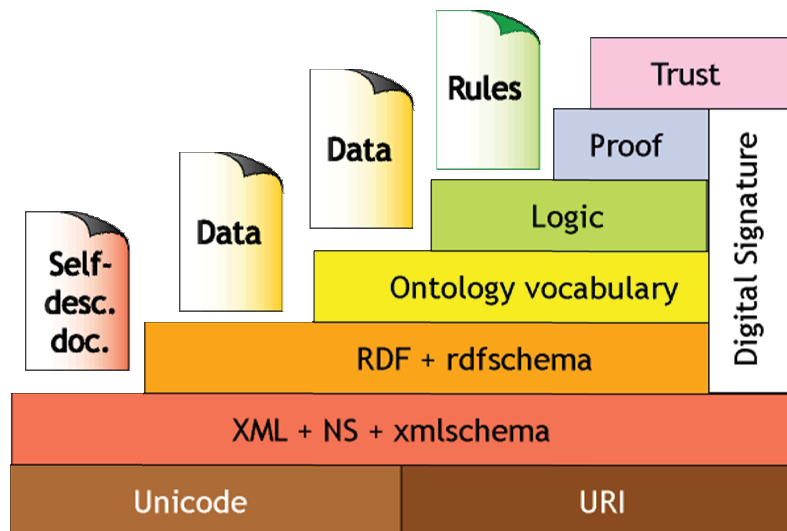


Figure 14: Layer approach of the Semantic Web [Antoniou and Van Harmelen, 2008]

As seen previously, Semantic Web is based on ontology. *Ontos* (from greek) means *being* and *logia* means *science, study, theory*. Ontology formally and explicitly classifies and describes concepts within a specific domain of knowledge [Batsakis and Petrakis, 2011] with axioms, definitions and theorems [Bittner et al., 2006]. In addition, knowledge can be stored and exchanged over a worldwide network with ontologies [Fortineau et al., 2013]. An interesting definition is proposed by Gruber [Gruber, 1993]: “An ontology is an

explicit specification of conceptualization. It is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent". Another definition is from Uschold [Uschold and Gruninger, 1996]: "An ontology is a shared understanding of some domain of interest". And finally the definition of Guarino [Guarino, 1998] is: "An ontology represents an engineering artifact, composed of a specific vocabulary to describe a certain reality".

Having many advantages, ontology has been applied to other domains such as design and manufacturing [Zhong et al., 2013]. Moreover a common vocabulary between all stakeholders is provided by ontology [Lee and Jeong, 2012]. Ontological models have been introduced and developed to formally describe domains of knowledge. In a recent past, research efforts have been made in the knowledge representation area in order to propose ontological models of the actual world in a machine-interpretable manner. The ontology allows formalizing the modeling concept, with axioms, definitions and theorems [Bittner et al., 2006] and provides assistance in modeling its dynamic behavior [Horrocks and Sattler, 2001]. As such, ontology is not only a database.

Ontologies were initially introduced for the formalization of rules, by using the various classes connected to each other with related object properties. Moreover formal ontology is also developed for checking models consistency, especially with logics and semantics [Baader et al., 2007]. Among the related features, ontology enables the reuse of domain knowledge, the explicit making of domain assumptions and the distinction of domain knowledge from the operational knowledge.

For instance Kitamura et al. [Kitamura et al., 2004] have used ontology to capture and to reuse knowledge about product functions in an electric company. As such, one of the key features is to make explicit the designer's knowledge, which has not yet been recognized by any best practices. So with such ontology development, knowledge sharing is enabled [Kusiak and Salustri, 2007]. Inference ontology is expressed with the OWL and DL languages [Fortineau et al., 2013] and aids users' activities by automatizing few of them. It also uses logical deductions on information and ensures the information consistency of the product along its lifecycle.

There are three different types of OWL languages: OWL-Lite, OWL-DL and OWL-Full. OWL-Lite is a smaller subset of OWL-DL and is easier to implement. OWL-DL has a high expressivity and is decidable (all started computation will be finished at a given time). OWL-Full has the highest expressivity, but is undecidable.

### **2.3.2 Product design rationale, intents and formalisms**

Product design intents represent all the mental activities of a designer during the product development process. It describes the motivation, rules and reasons of the designer's decisions [Liu and Sun, 2008]. The capture (by reasoning) and the representation of design intents can be separated into three categories in order to be shared with collaborators [Kim et al., 2006]: function, feature and **Design Rationale (DR)**. The latter lists all de-

cisions made during the design process and explains why an artifact is designed the way it is [Zhang et al., 2013].

Retrieving DR is considered as a useful and accurate way of capturing and representing relevant hidden knowledge in product design models [Wang et al., 2012]. DR therefore provides good support for redesign, design reuse, maintenance, evaluation and verification to name a few [Chandrasegaran et al., 2013] in either informal, or semiformal or formal manner where computational capabilities are still needed [Bracewell et al., 2009]. Furthermore, the current lack of engineering knowledge in product design highlights needs in managing DR in a more appropriate manner [Chandrasegaran et al., 2013] by using specific formalisms such as **Core Product Model (CPM)** and so on.

### 2.3.3 Spatiotemporal ontologies

**SUMO (Suggested Upper Merged Ontology)**, owned by IEEE, has originally been built by Teknowledge Corporation [Pease, 2014]. The ontology is composed of meta-level concepts (such as mereotopology, processes and objects), mid-level ontology and lots of domain ontology (cf. Figure 15). Actually, it is the largest formal public ontology with 25,000 terms in domains such as engineering components, geography, justice, shopping and so on. It is mapped to the entire WorldNet (freely available online database) lexicon. Its goal is to have a consistent and machine-interpretable model, which can interoperate and reason (using Sigma) between systems. Its hierarchy tree is composed of physical and abstract entities, highlighted in grey. Physical entities (i.e. things which have a position in space/time) are for instance object (i.e. region) and process (i.e. motion). Abstract entities (i.e. things which do not have a position) are for example spatial and temporal relations.

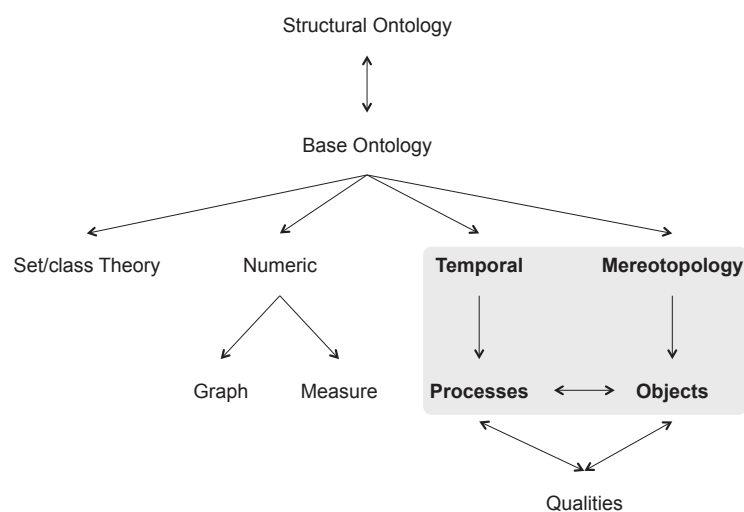


Figure 15: Hierarchy of the SUMO ontology [Pease, 2014]



In mechanical engineering field, Kim et al. [Kim et al., 2006] have developed an assembly design ontology and an ontology-based assembly design framework, by using mereotopological primitives and SWRL formalism [Kim et al., 2008], in order to facilitate collaborative product development. In this framework, design intents can be well understood among different designers, and applications can reason about assembly knowledge without any semantic ambiguity. Based on these preliminary efforts, Demoly et al. [Demoly et al., 2012b], with PRONOIA, described product relationships with mereotopological primitives and have implemented their descriptions into an ontology with OWL-DL and SWRL languages, so as to be machine-interpretable by PLM systems. Design ontologies can guide the system users in understanding and applying knowledge carried by the system [Horvath et al., 1998]. From a PLM point of view, the ontological applications are mostly carried out in the BOL (i.e. including design and manufacturing phases). On the contrary, only few studies are addressed at the MOL (i.e. including usage and maintenance phases) [Matsokis and Kiritsis, 2010].

Furthermore, the **NIST** (**N**ational **I**nstitute of **S**tandards and **T**echnology) has developed CPM to support PLM information [Fenves et al., 2008b] (e.g. functional reasoning about a product in the conceptual design stages).

Moreover Matsokis et al. have developed an ontology model of the product data and knowledge management **Semantic Object Model (SOM)** [Kiritsis et al., 2003] using UML class diagram. An efficient description of the product as it is designed from the manufacturer point of view and a functional structure for storing data of the product lifecycle is developed in SOM [Matsokis and Kiritsis, 2010]. So it improves interoperability in PLM.

Additional efforts have been made to combine space and time so as to represent product evolution. For instance, Welty et al. [Welty and Fikes, 2006] have developed an ontology using *fluents*, allowing objects to support properties' changes over time. As such, objects are composed of *timeslices* (i.e. temporal parts) and *fluents* are only valid during a specific interval.

In addition an European project called **TOWL** (**T**ime-determined **O**ntology **W**eb **L**anguage) aims at extending an extended OWL language with a temporal dimension [Viorel, 2007]. Finally Hobbs and Pan [Hobbs and Pan, 2004] propose a temporal ontology, called OWL-Time, so as to describe the temporal properties of web services using Allen's intervals.

As part of formal ontology, qualitative reasoning can be applied either in spatial dimension [Duntsch et al., 2001], temporal dimension or both, spatiotemporal [Gereveni and Nebel, 2002]. One of the main pillars of qualitative reasoning is **Qualitative Spatial Reasoning (QSR)**, which represents and reasons with spatial entities of higher dimension [Aiello et al., 2007]. On the other hand, some applications use qualitative temporal information to specify temporal relationships between activities (e.g. one task has to be carried out after another one) [Schockaert et al., 2007].

From a PLM point of view, the current efforts in ontology-based approaches are to translate existing models and also to develop new models into ontologies. The related ontological applications are mostly addressed in the BOL (including design and manufacturing phases), but only few studies are carried out at the MOL (including usage and maintenance phases) [Matsokis and Kiritsis, 2010].

## 2.3.4 Main ontology models in product design

### 2.3.4.1 FBS model

**FBS** (**F**unction-**B**ehaviour-**S**tructure) framework has been first introduced by Gero [Gero, 1990] and been described in an ontology [Gero and Kannengiesser, 2004]. It was originally focusing on artificial objects. FBS is nowadays used to represent and classify processes. FBS provides information on:

- Function (F) of an object is defined as its teleology (“what the object is for”);
- Behaviour (B) of an object is defined as the attributes that can be derived from its structure (“what the object does”);
- Structure (S) of an object is defined as its components and their relationships (“what the object consists of”).

As such, three variables define the design conceptual state space [Colombo et al., 2007]: function variables (F), expected behavior variables (Be) and structure variables (S) (cf. Figure 16). The design activity is aimed at translating functional requirements into structures that are able to realize them. FBS model is generally suggested as a way to understand the typical life cycle of a decision making process in design.

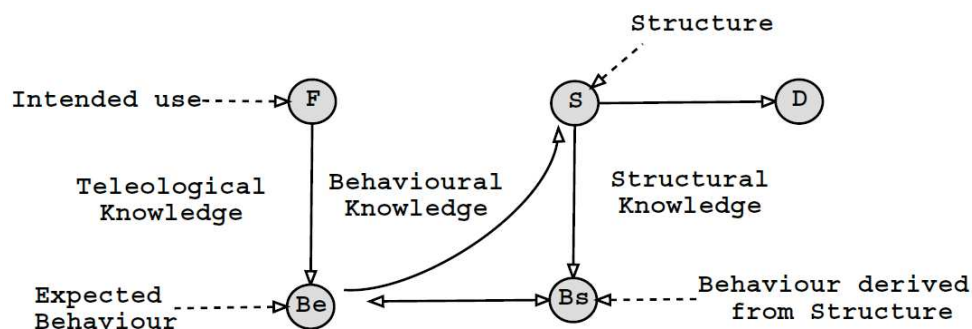


Figure 16: FBS model under the viewpoint of the engineering design knowledge [Colombo et al., 2007]



### 2.3.4.2 PRONOIA approach

After defining product relationships based on mereotopological theory, the approach called PRONOIA has implemented the theory into an ontology in order to reuse the information in other lifecycle phases [Demoly et al., 2012d]. The authors focus on the information exchange between product design and assembly process phases. The ontology is implemented in Protégé Editor using OWL-DL and SWRL languages. As such, a further insight of assembly and design intents is provided to the product architect. Moreover, the qualitative information is available for reuse in MOL and EOL. The proposed PRONOIA model is shown in Figure 17 [Demoly et al., 2012b].

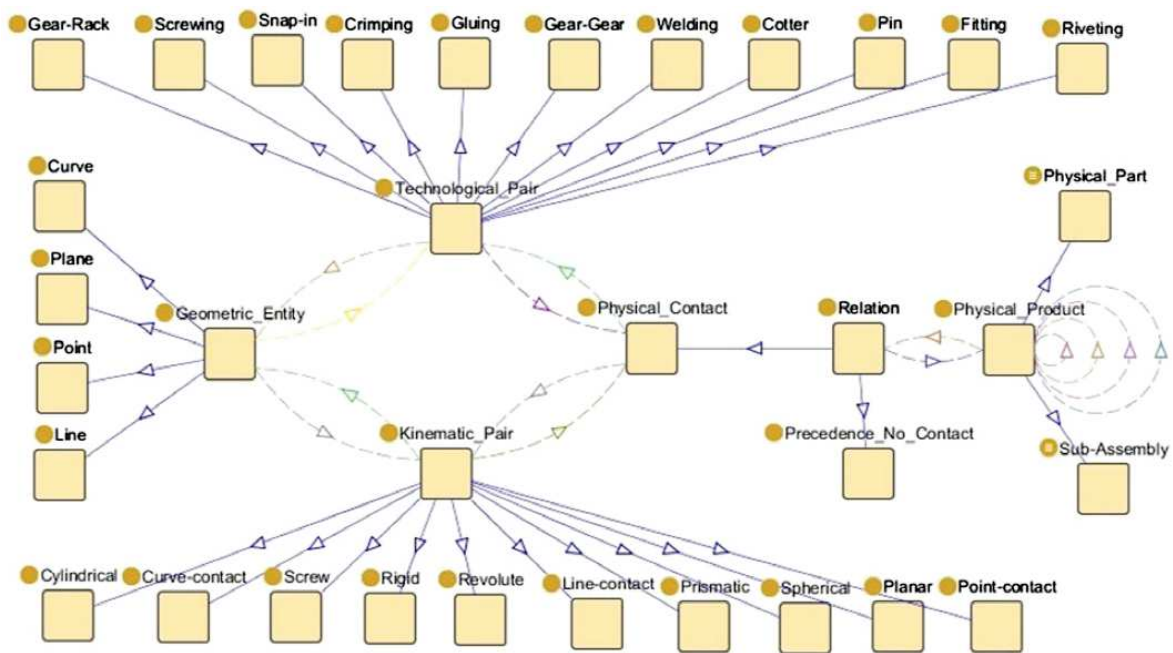


Figure 17: The PRONOIA ontology model in Protégé [Demoly et al., 2012b]

The model has been developed to describe the assembly properties as they have been defined with the mereotopological theory. The PRONOIA approach is particularly relevant in product engineering where relevant information is mainly embedded in CAD models and so is not enough exploited. Table 6 presents some of the description logic restrictions, which are used to reason on this ontology model.

### 2.3.4.3 CPM and OAM models

CPM is a data model created to support the PLM along the product life: from the product's conceptualization to its disposal. It is based on three artifacts: Function (i.e. describes what the artifact is supposed to do), Form (i.e. represents the proposed design solution for the design problem specified by the function) and Behavior (i.e. describes how the artifact's form implements its function) [Fenves et al., 2008b]. Fenves begins to

Table 6: Some of the description logic restrictions of PRONOA [Demoly et al., 2012b]

Class name	Rule type	DL Rule
Geometric_Entity	Cardinality	$\text{Geometric\_Entity} \equiv 1 \text{ is\_Geometric\_Entity\_Of2Kinematic\_Pair}$
Kinematic_Pair	Cardinality	$\text{Kinematic\_Pair} \equiv 1 \text{ is\_Kinematic\_Pair\_of2Physical\_Contact}$
Relation	Cardinality	$\text{Relation} \equiv 2 \text{ Relation2Physical\_Product}$
Physical_Contact	Cardinality (inherited)	$\text{Physical\_Contact} \equiv \text{Relation} \sqcap = 2 \text{ Relation2Physical\_Product}$
Revolute	Cardinality	$\text{Physical\_Contact} \equiv \text{Kinematic\_Pair} \sqcap = 1 \text{ has2Line}$
	All Values From	$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap \forall \text{Kinematic\_Pair\_has2Geometric\_Entity} . (\text{Line} \sqcup \text{Plane})$
	All Values From	$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap \forall \text{Kinematic\_Pair\_has2Line} . \text{Line}$
	Some Values From	$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap \exists \text{Kinematic\_Pair\_has2Line} . \text{Line}$
	Cardinality	$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap = 1 \text{ Kinematic\_Pair\_has2Line}$
	All Values From	$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap \forall \text{Kinematic\_Pair\_has2Plane} . \text{Plane}$
	Some Values From	$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap \exists \text{Kinematic\_Pair\_has2Plane} . \text{Plane}$
	Cardinality	$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap = 1 \text{ Kinematic\_Pair\_has2Plane}$
	Cardinality (inherited)	$\text{Revolute} \equiv \text{Kinematic\_Pair} \sqcap = 1 \text{ is\_Kinematic\_Pair\_of2Physical\_Contact}$
	Cylindrical	Some Values From
All Values From		$\text{Cylindrical} \equiv \text{Kinematic\_Pair} \sqcap \forall \text{Kinematic\_Pair\_has2Line} . \text{Line}$
Cardinality		$\text{Cylindrical} \equiv \text{Kinematic\_Pair} \sqcap = 1 \text{ Kinematic\_Pair\_has2Line}$
Cardinality (inherited)		$\text{Cylindrical} \equiv \text{Kinematic\_Pair} \sqcap = 1 \text{ is\_Kinematic\_Pair\_of2Physical\_Contact}$
Sub-Assembly	Min Cardinality	$\text{Sub\_Assembly} \equiv \text{Physical\_Product} \sqcap \geq 2 \text{ is\_Parent\_Of}$
Physical_Part	Complement Of	$\text{Physical\_Part} \equiv \text{Physical\_Product} \sqcap \neg \text{Sub\_Assembly}$

use object-oriented languages (e.g. **UML** – **Unified Modeling Language**) to represent the product. CPM has been converted to the OWL language later on (cf. Figure 18). The classes in CPM are grouped into the following four categories [Fiorentini et al., 2010]:

- Classes that provide supporting information for the objects (CoreProductModel, CommonCoreObject, CommonCoreRelationship, etc.);
- Physical or conceptual objects classes (such as Feature, Requirement, Function);
- Classes that describe relationships among the objects (Constraint, EntityAssociation, Usage);
- Classes that are commonly used by other classes (Information, ProcessInformation, and Rationale).

Then CPM has been extended to cover assembly issues through **OAM** (**Open Assembly Model**), a standard representation and exchange protocol for assemblies. In fact, a generic product can be represented by CPM and OAM specially represents mechanical assemblies. OAM introduces information such as how the component is composed and what are the assembly relationships. For instance, OAM considers that an assembly is a composition of its subassemblies and parts. Both *assembly* and *part* classes are sub-classes of the CPM artefact class [Fiorentini et al., 2010]. The OWL version of OAM is represented on Figure 19.

In addition, both CPM and OAM have been transformed to an inference model, called CPM2, which supports a broad range of information relevant to PLM [Fenves et al., 2008a].



#### 2.3.4.4 AsD approach

Kim et al. [Kim et al., 2004] proposed a spatial relationships-based assembly design formalism describing assembly relations, and an assembly relation model, in which relations were represented in XML format. This first attempt was only focused on the geometric aspect of the product and did not enable an efficient interpretation by CAD applications. Recently, Kim et al. [Kim et al., 2006] have tried to demonstrate the feasibility of an ontological representation of assembly and associated constraints. As such, they have introduced the description in the **AsD** (Assembly Design) ontology [Kim et al., 2008], after describing assembly joints with a mereotopological theory.

The main purpose of this system is to support designers during the product assembly and joint design process [Kim et al., 2004]. As such, the product achieves high performance in its whole lifecycle. The assembly knowledge is formally and explicitly specified with the ontology. It is also machine-interpretable and can be shared among various information systems, so as to promote collaborative assembly information-sharing environments. A standardized data format is actually required to capture designer's intents and the semantics of the designer's terms. Moreover, the AsD ontology captures a DR, including joint intent and spatial relationships. Protégé Editor 3.1 has been used to generate the ontology with the OWL and SWRL language [Kim et al., 2006]. Figure 20 illustrates the hierarchy of Assembly Design ontology classes.

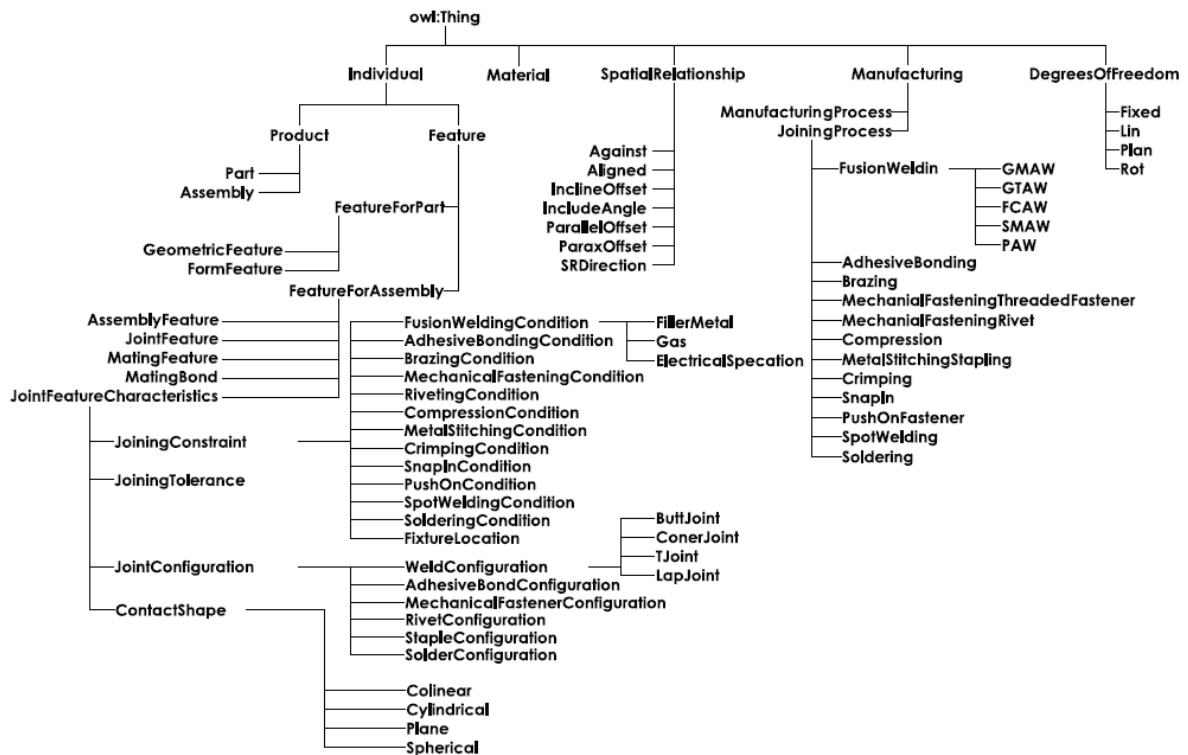


Figure 20: Assembly Design ontology class hierarchy [Kim et al., 2008]

Table 7: Some of the SWRL rules [Kim et al., 2008]

	SWRL rules
Overlap	$individual(?x) \wedge individual(?y) \wedge individual(?z) \wedge is\_part\_of(?z, ?x) \wedge is\_part\_of(?z, ?y) \rightarrow overlap(?x, ?y)$
Cross	$individual(?x) \wedge individual(?y) \wedge individual(?p) \wedge individual(?q) \wedge is\_part\_of(?x, ?p) \wedge differentFrom(?p, ?y) \wedge disjoint(?x, ?q) \wedge differentFrom(?q, ?y) \rightarrow cross(?x, ?y)$

SWRL rules are used to represent and reason on the differences between joints and to define assembly design terms and their relationships. As such, assembly constraints are explicitly represented and differentiated with the reasoner. Table 7 sums up some of the SWRL rules [Kim et al., 2008].

### 2.3.5 Summary and position

Built on previous information, a map of existing ontologies is introduced in Figure 21 and classifies ontologies according to product and process domains and dimensions. Therefore, the proposed ontological development called PRONOIA2, extension of PRONOIA [Demoly et al., 2012b], aims at describing product and also process in the three dimensions (i.e. spatial, temporal and spatiotemporal) in order to describe the evolution of the product in AOD. This ontology model is composed of a meta-ontology and the domain-ontology so as to cover product and process information in the BOL.

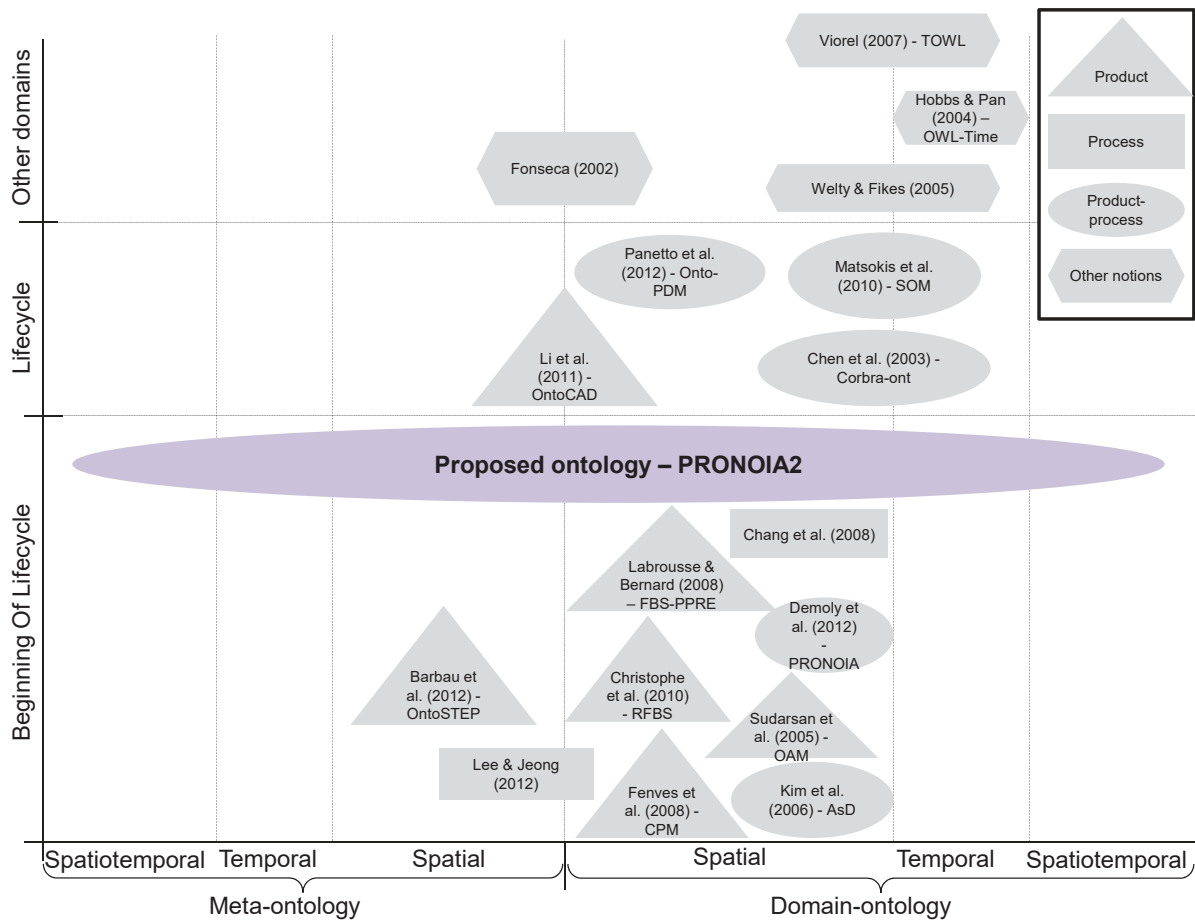


Figure 21: Map of existing ontologies along lifecycle phases [Gruhler et al., 2015a]

## 2.4 Product Lifecycle Management

### 2.4.1 PLM definition and goal

In the past, transfer information was only done manually on paper. This process was slow and mistakes often arose [Stark, 2006]. At that time, company has major issues to get real control of their products throughout their lifecycle. To enable such a control, the concept of PLM appeared in the 1990s [Kubler et al., 2015]. A definition for PLM is “a strategic business approach that applies a consistent set of business solution in support of the collaborative creation, management, dissemination and use of product definition information across the extended enterprise from concept to end of life – integrating people, processes, business systems and information”. As such, PLM can improve the communication between stakeholders working in a collaborative project, so as to aid them in their activities. Moreover, information flow and especially engineering activities are currently managed by PLM systems [Franke et al., 2011]. PLM deals with several type of information such as technical data, knowledge on constraints and requirements [David and Rowe, 2015]. Files traceability is ensured by PLM. Only files versions are currently managed. PLM does not



entirely support information about product changes.

Moreover, a PLM issue concerns the transfer of decisions from upstream phases to downstream phases and from downstream phases to design phases. Indeed in a “classical” design process, the designer has no information access about the constraints of the different lifecycle phases. Unfortunately in concurrent engineering, these constraints (e.g. assembly process constraints) have to be defined simultaneously with the design phase [Chungoora et al., 2013].

PLM is composed of different information systems. PDM is one of the components of PLM. It manages technical data such as eBOM, subassemblies, parts, product structures, documents, configurations, and provides comfortable support for designers and product architects. PDM aids in the product development process by giving the right information, at the right time, at the right person, in the right format. It also keeps track of the different iterations of the project files. However, PDM does not provide change management capabilities enabling seeing for instance the BOM before and after changes [Siemens Industry Software, 2015].

MPM system, which is also an important component of PLM, enables the management of manufacturing/assembly data such as mBOM, operations, documents and so on, and provides assistance in assembly planning and assembly line balancing phases. Based on this statement, a lack of associativity in PLM systems (especially between PDM and MPM) is highlighted [Demoly et al., 2012b].

David and Rowe [David and Rowe, 2015] go further and state that PLM systems need a complex ontological model to process and integrate such information diversity. In fact, many researchers are working on the organization in a single integrated system linking PLM systems in order to avoid heterogeneity issues [Fortineau et al., 2013].

## **2.4.2 Engineering change management**

Throughout the product development process, many versions of files are created. Indeed, products are developed with different options and alternatives. As such, each time a design change is done, the files are iterated or a new version is created. Therefore management tools to support changes and modifications are required in order to maintain information flow control through the product lifecycle [Stark, 2006].

### **2.4.2.1 Changes**

An engineering change is an alteration made to any parts, drawings or software that has already been released during the product design process [Quintana et al., 2012]. Changes can be caused by several reasons. The draft standard ISO 11442-6 presents the following examples [Pikosz and Malmqvist, 1998]:

- 
- Change of a part depending on altered function or production requirements;
  - Change in the application of a part;
  - Introduction of a new part;
  - Replacement of a part;
  - Withdrawal of a part;
  - Correction of errors on a document;
  - Bringing an old document up-to-date.

Hoffman and Joan-Arinyo [Hoffmann and Joan-Arinyo, 2000] define the event which implies changes. His classification focuses on elements (or entities), but not on the associations between these elements:

- The element has moved;
- The element has been deleted;
- The element has been joined with another element;
- The element has been split into several new elements;
- The element has been enlarged or restricted;
- The element has been created.

The product is continuously changing throughout its lifecycle. Changes need to be known and understood at each development phase by each stakeholder. In fact, stakeholders contribute to the product evolution [Louhichi and Rivest, 2014] by adding information on the “original” product. Product development is also an iterative process. Making early design decisions has benefits but often requires modifications or engineering changes [Smith and Eppinger, 1997]. That is the reason why changes must be managed at the early design stages in order to always control product changes and have a consistent design [Fei et al., 2010].

As changes happen throughout the product lifecycle, they need to be managed in an effective manner. **ECM** (**E**ngineering **C**hange **M**anagement) is a key element of product development process [Parametric Technology Corporation, 2006]. Indeed, the fact of managing changes will increase stakeholders’ knowledge on the product as they will better understand product and process evolutions [Hamilton, 2010].



### 2.4.2.2 Definition and needs of ECM

ECM control changes workflow in order to keep traceability on who did what and when. Changes proposition, analysis, planning, implementation and validation are managed throughout the product evolution [Parametric Technology Corporation, 2006]. The main research work in the literature is focused on changes management when changes occur during the manufacturing phases. The impact of these changes on the product and on the delivered products is also studied [Weber, 2015]. The necessity to manage changes and product configurations is directly proportional to the product complexity [Parametric Technology Corporation, 2006].

In industry, ECM is recognized as a problem that receives too little attention relative to its importance [Tavcar and Duhovnik, 2005]. Indeed, ECM processes are difficult to be applied in companies [Parametric Technology Corporation, 2006]. This process becomes harder and harder when different data management systems (e.g. PDM and MPM) are used by the enterprise [Parametric Technology Corporation, 2006]. Current PDM and MPM systems miss the management of change propagation [Rouibah and Caskey, 2003]. However, changes should be communicated to officially design a product. As such, stakeholders will be aware of what is going to happen, when and why [Stark, 2006]. A solution is to improve the manufacturing change process and create an environment that encourages a strong relationship between the engineering and manufacturing departments [Arenas Solutions, 2015].

Therefore, product evolution will be controlled and designers' understanding will be improved. The best way to achieve ECM is to apply it as early as possible in the lifecycle process so that changes do not impact the whole product and expensive costs can be avoided. In an unfortunate case where designers misunderstand changes management process, the consequences will be [Parametric Technology Corporation, 2006]:

- Incomplete documentation;
- Loss of design history;
- Issues between development process phases.

Competitiveness is largely improved with the use of ECM. The major advantages are [Parametric Technology Corporation, 2006]:

- Unique enterprise process (e.g. a hub manages and gathers information);
- Better tasks coordination with the automation process;
- Product costs reduction ;
- Product quality improvement ;
- Time-to-market reduction.

Besides, ECM can provide information sharing, simultaneous data access and prompt communication [Huang et al., 2001]. ECM process can ensure stakeholders earn 33% of time compared to a “classical” product development process [Parametric Technology Corporation, 2006].

### 2.4.3 Main engineering approaches to maintain consistency in product design and BIM

#### 2.4.3.1 Demoly’s et al. approach

Demoly et al. have developed a novel approach to integrate assembly process information and knowledge in the early phases of the product development process. They are working in the domain of AOD. Their proposed approach – called **SKeLeton** geometry-based **Assembly Context Definition (SKL-ACD)** – enables the control of the product modeling phase by introducing skeleton entities consistent with the product relationships and the assembly sequence planning information [Demoly et al., 2011a]. Figure 22 introduces the process from the directed graph composed of kinematic pairs to the skeleton graph.

#### 2.4.3.2 Louhichi’s and Rivest’s approach

The objective of this approach is to maintain consistency between CAD work packages and the global product **Digital Mock-Up (DMU)**. Louhichi and Rivest [Louhichi and Rivest, 2014] want to control the DMU evolution, as well as the modification undergone. The approach is designed to ensure better management of the associations between objects. Figure 23 presents how the approach determines the associations between CAD elements. Here associations are not found, as the product sides are different after modification. Besides, collaborative work is supported between project stakeholders, as information sharing is facilitated.

#### 2.4.3.3 Chen’s and Luo’s approach

Chen and Luo [Chen and Luo, 2014] are working on making consistent the information travelling from design to construction phases. Indeed, transformation of processes can be supported by **BIM (Building Information Modeling)**. The advantages are:

- Visualization and coordination of **AEC (Architecture, Engineering and Construction)**;
- Less errors and omissions;
- Improvement of the productivity;
- Support for scheduling.

BIM has attempted to manage spatial and temporal information (cf. Figure 24). 3D spatial (i.e. 3D product) and 1D temporal (i.e. construction activities) information is supported during the development process of a construction project [Issa et al., 2003]. BIM also aids stakeholders to schedule activities in a project and manage changes [Pooyan et al.,

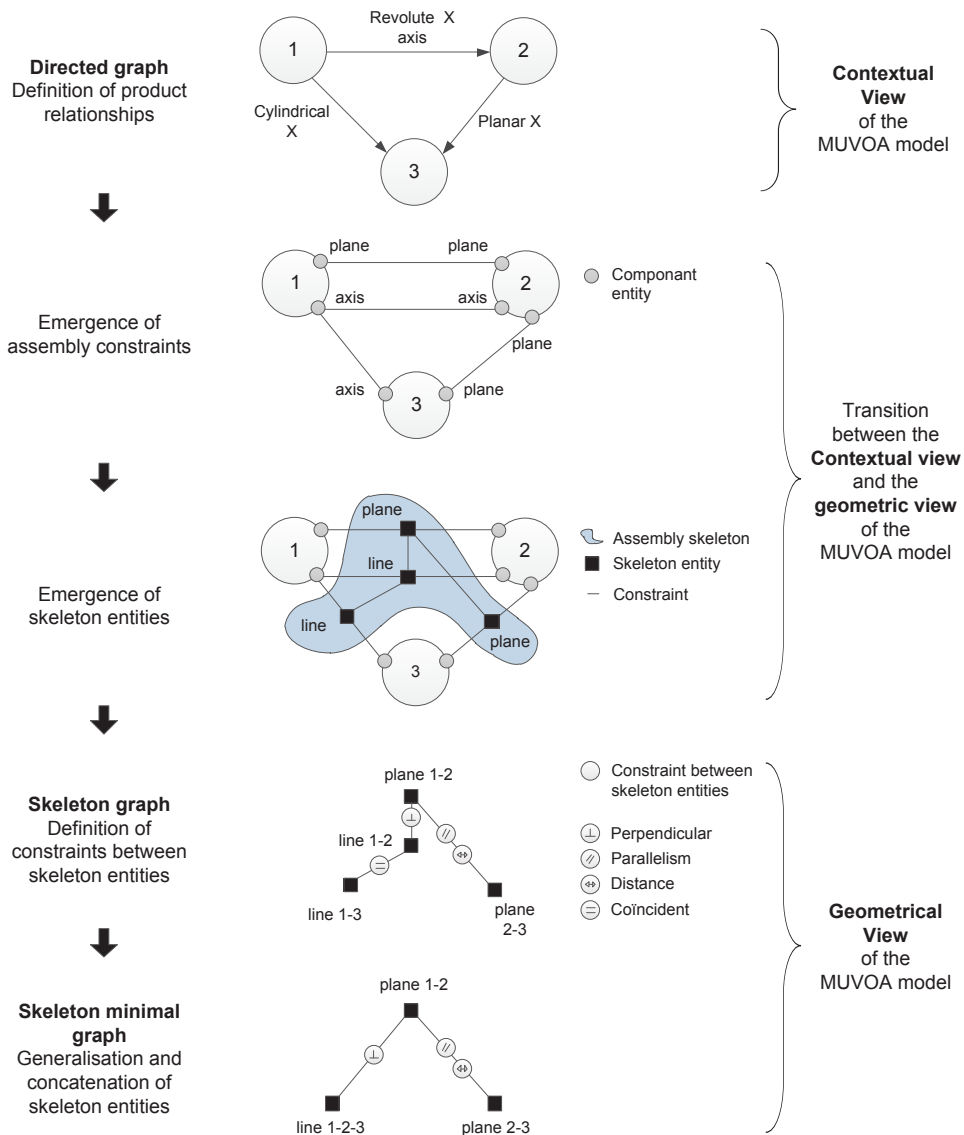


Figure 22: Definition process of a minimal skeleton model [Demoly et al., 2011a]

2009]. A planning is generally added to current design tools in the form of a time sequence. So the model is constructed with the following data structure: product, organization and process. Contrary to spatial and temporal information, here spatiotemporal dimension is not managed.

### 2.4.4 Heterogeneity management in other domains

Heterogeneity is a recurrent issue, encountered in projects from various domains. Indeed, GIS often deals with this kind of issues. For instance, Straume [Straume, 2014] works with

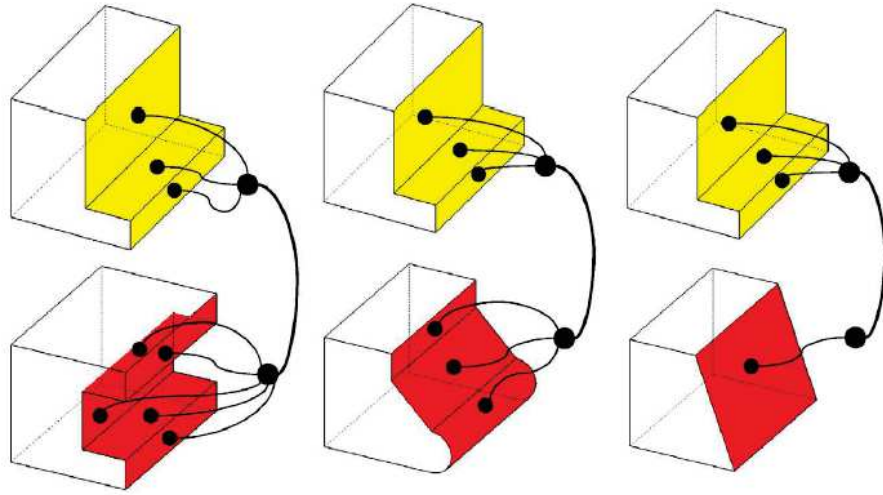


Figure 23: Determination of association between CAD elements [Louhichi and Rivest, 2014]

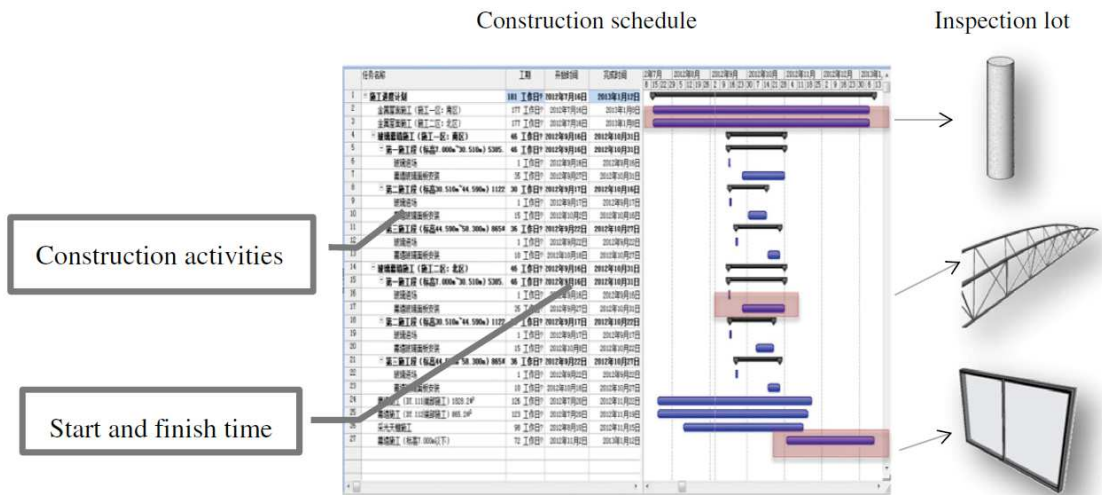


Figure 24: Relationship between construction activities and BIM components [Chen and Luo, 2014]

land management tools to have a consistent product over space and time. In fact, stakeholders do not have the same vocabulary to describe the project evolution. As such, land changes over time can be identified and team partner can work collaboratively. Moreover, Fonseca et al. [Fonseca et al., 2000] use ontology to enable interoperability, which is the ability of a system to share information.

Heterogeneous issues also occur between two research domains. For instance, the reduction of the heterogeneity gap between GIS and BIM has been studied by Mignard

& Nicolle [Mignard and Nicholle, 2014]. A platform (called ACTIVE3D) and an ontology are used to standardize the knowledge representation relating to building and geographic objects. Similar works, developed by Pittet et al. [Pittet et al., 2014], try to make the information exchange representation homogeneous between stakeholders in the domain of **FM** projects (**F**acility **M**anagement) and BIM. An ontology is built to organize and structure knowledge produced by each stakeholder during the building lifecycle. In the construction sector, Rezgui et al. [Rezgui et al., 2011] use ontology to support processes and make them automatic. In engineering domain, CAD and GIS are being integrated by Peachavanish et al. [Peachavanish et al., 2006]. This enables interoperability between GIS (to perform location related analysis of components at different scales) and CAD (to capture and represent information about components) information. This research work enables supporting engineering and construction, as well as avoiding a waste of time and money due to no interoperability.

### **2.4.5 Summary and position**

Based on previous statement, the literature has highlighted the need to manage changes during the early development process of the product. The fact of integrating assembly constraints during design phases is especially useful. Figure 25 classifies management approaches according to the dimension of the managed entities and the temporal position in product lifecycle. Besides, a lack of spatiotemporal information (i.e. changes) management is highlighted. Main approaches concern the management of information in the spatial dimension and few of them in the temporal dimension. The proposed approach will manage spatial, temporal and spatiotemporal information in the context of AOD so as to aid designers' and product architects' understanding of product evolution.

## **2.5 Summary: need of theory, model and approach**

A state of the art, on mereotopological theories, ontologies and information management approaches, has been carried out. A comparison has been made in each domain to highlight the most suitable theory, ontology and approach. These three domains of research are not disjoint, but complementary. In fact, a lonely mereotopological theory cannot be useful if it is not related to an ontology, which can formalize it [Kim et al., 2006]. The integration of the theory in an editor makes designers and product architects able to apply it. In the same way, the ontology can check the information consistency and bring a common and structured semantic [Matsokis and Kiritsis, 2010]. Moreover, an articulation between information systems (i.e. PDM and MPM) is required so as to ensure a consistent information flow between design and assembly phases. The approach will also improve product architects and designers' understanding and awareness [Card et al., 1999]. As such, each of these three domains adds further information, which will be helpful to project's stakeholders.

These three domains also enable seeing the design under a novel paradigm: instead of having an endurantist product development only based on spatial dimension, the approach

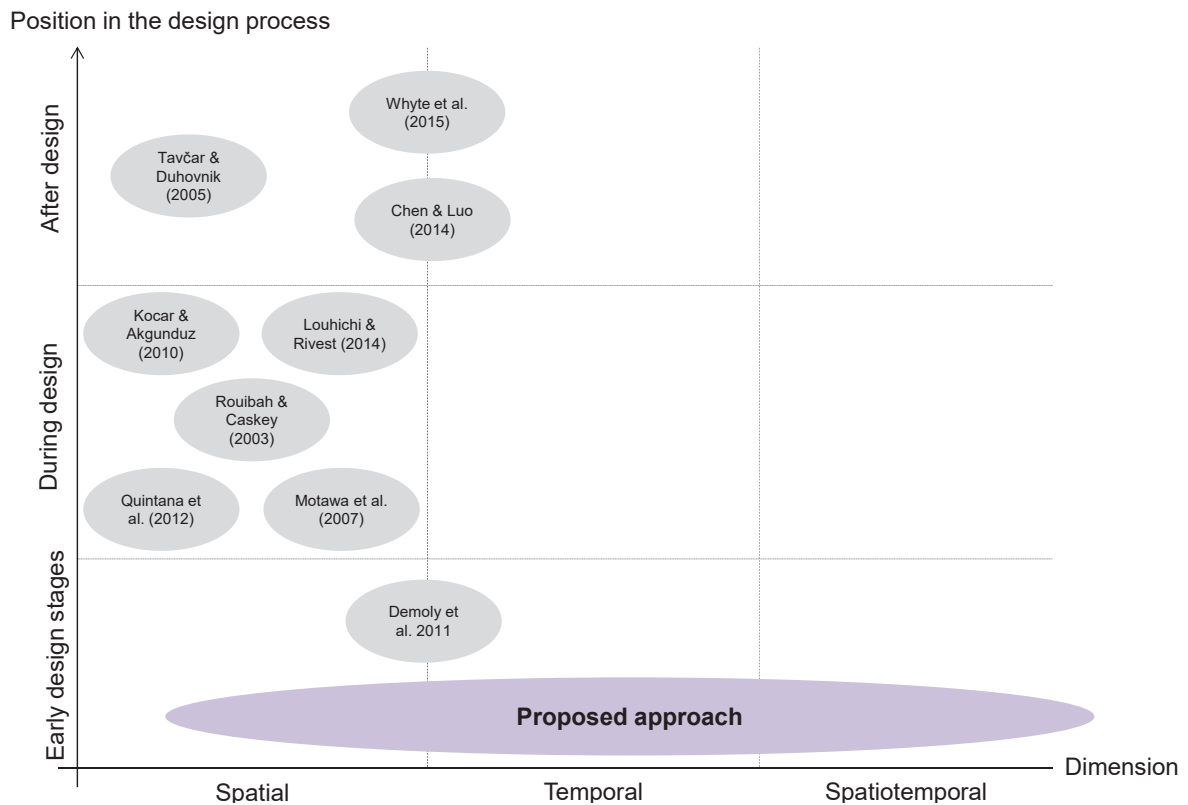


Figure 25: Map of existing management approaches

that we propose is based on three dimensions. Current design is indeed only composed of mechanical parts and assemblies (as seen in CAD tools). On the contrary, our novel principle is based on the description of product in the three dimensions: spatial (i.e. for all information concerning the product), temporal (i.e. for all information concerning the process) and the introduction of the spatiotemporal dimension [Zeng and Gu, 1999b]. These three dimensions enable the description of the product evolution in the context of AOD with the “engineering sense” [Salustri, 2002].

This novel approach enables the management of knowledge (i.e. the mereotopological theory formalizes implicit knowledge coming from product architects) and information (i.e. the theory is implemented in an ontology to be machine-interpretable and the approach manages the information flow through the different systems).

As such, our research work is based on a theory (the mereotopological theory), a model (the ontology) and an approach (the framework and information flow). The problematic needs theory, model and approach to be solved. They will be described in details in the next chapters.



# Chapter 3

## Spatiotemporal mereotopology-based theory for product-process description – JANUS

### Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>52</b>
<b>3.2</b>	<b>Overall description of JANUS theory</b>	<b>53</b>
3.2.1	General philosophy	53
3.2.2	Object change description	54
<b>3.3</b>	<b>Description of the spatial dimension of JANUS</b>	<b>55</b>
<b>3.4</b>	<b>Description of the temporal description of JANUS</b>	<b>57</b>
<b>3.5</b>	<b>Description of the spatiotemporal description of JANUS</b>	<b>58</b>
3.5.1	Basic problems with spatiotemporal visualization	58
3.5.2	Filiation relationships	61
3.5.3	Spatiotemporal primitives classification	61
3.5.4	Kinematic pairs and Design changes description	62
3.5.5	Technological pairs description	64
<b>3.6</b>	<b>Summary</b>	<b>76</b>

---



### 3.1 Introduction

In this chapter, the major objective is to propose and describe a region-based theory called mereotopology for AOD process to describe products evolution. With such a theory, the formal representation of the product relational information with its parts and its lifecycle operations (i.e. here assembly operations) at various abstraction levels, is a critical point in the product design process (Figure 26). At this stage, an additional layer is needed so as to ensure a seamless integration of assembly sequence planning in product design with semantics and logics.

Thus, spatial objects as well as temporal objects are introduced in order to formally describe product-process relationships in a understandable and machine-interpretable manner. The fact of introducing spatiotemporal objects (such as swept volumes) and spatiotemporal relationships enables the appropriate understanding of spatial objects evolution over time and then their associations. By introducing such a formal representation, AOD tackles current engineering issues related to the integration of a product lifecycle constraints and knowledge in a proactive manner. The paradigm of perdurantism leads product architects and designers to play a role through spatial, temporal and spatiotemporal dimensions (also considered as four-dimensionality) [Sider, 2001].

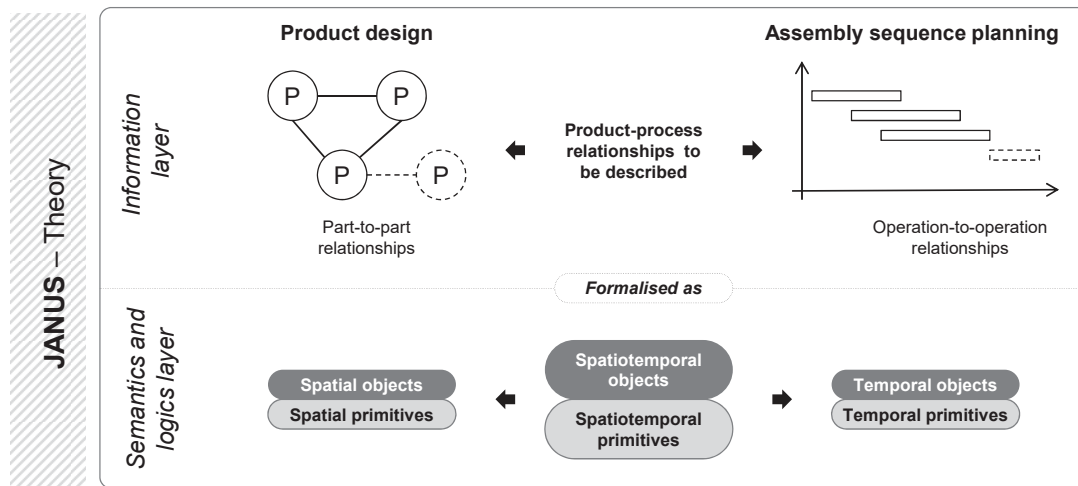


Figure 26: Introduction of a semantics and logics layer for ensuring product-process integration

The current chapter describes the proposed theory, which is called JANUS, in the field of AOD, and particularly focused on integrated product design and assembly sequence planning [Gruhier et al., 2014c]. The theory foundation (described in the second paragraph), based on the related objects and primitives is defined in the spatial (cf. third paragraph), temporal (cf. fourth paragraph) and spatiotemporal (cf. fifth paragraph) dimensions. At the end, the main strengths of JANUS theory are summed up.

## 3.2 Overall description of JANUS theory

### 3.2.1 General philosophy

The proposed theory foundation is inspired from Bergson’s statement [Bergson, 1923], in which three temporal situations are considered, such as: the present (i.e. the existence), the past (i.e. the essence) and the future (i.e. the prediction) [Heidegger, 1962]. In this context, it is stated that the existence of an object precedes and leads to its essence: an object exists before being defined by concepts. The object cannot be first defined, as at the beginning the object does not exist (e.g. the assembly does not begin with all parts but just with few of them).

The first principle of existentialism [Sartre, 1946] states that the object is designed after the existence. The object exists only during its lifecycle when it is useful and in relation with others. It represents the same idea as the sentence from Descartes “I think, therefore I am”. The object is not present as long as it is not recognized by the others as useful (i.e. so until it is not in relation with others). The others are therefore its condition of existence [Sartre, 1946]. When the object is in relation with others, it realizes its function (i.e. the reason why this object has been designed). Without its function the object is useless and so does not exist [Sartre, 1946]. For instance, sometimes when parts, involved in the functional flow, are added or moved, functions of the entire product can change.

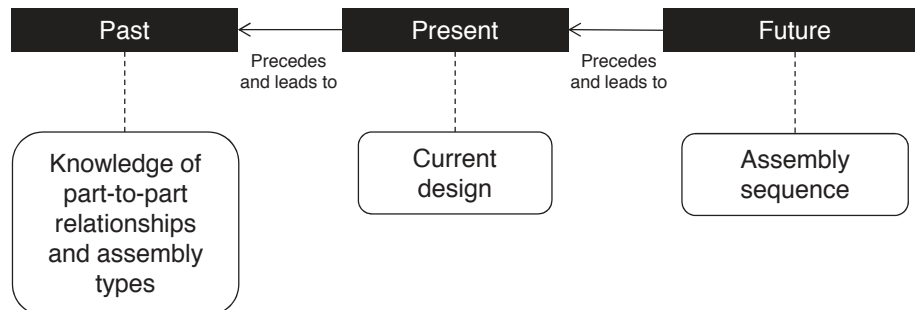


Figure 27: The three temporal situations of Bergson applied to assembly design on which the proposed theory is based [Gruhler et al., 2014e]

Figure 27 aims at introducing the novel philosophy on which the proposed theory is based. This figure highlights that the future precedes and leads to the present, and likewise the present precedes and leads to the past. The authors state that the three temporal situations of Bergson are adaptable to AOD. The purpose of the AOD approach is to consider the assembly sequence as early as possible in the product development process, so as to provide an assembly-oriented design context for designers. As such, by considering the assembly sequence and the assembly (i.e. part-to-part) relationships, product architects and designers can work with consistent product-process information knowledge and also

are aware of the temporal and spatiotemporal aspects of product-process definitions.

Moreover, the JANUS theory is derived from ISO 159262 Standard [Standard ISO 15926, 2003] – used in oil and gas industry – to describe a foundation based on industrial needs for supporting the information changes over time [Demoly et al., 2012b]. This standard is based on four-dimensionalism, which means that object can be extended over space and time [Batres et al., 2007]. In such a context, the object can be described in four dimensions so as to enable the existence of the past, present and future. The object is then considered as being perdurant over time and is defined according to its relationships with others. In this approach change is expressed using mereotopology through the object lifecycle [Sider, 2001].

### 3.2.2 Object change description

The theory frame, used to describe the object evolution during the AOD, is inspired from Le Moigne’s framework [Le Moigne, 1994]. By stating that a product is composed of spatial parts assembled over time, he has considered that objects differ from others according to:

- Their position in space (i.e. provide the localization compared to other objects);
- Their position in time (i.e. provide the temporal position of the object regarding to others);
- Their form (i.e. provide the object structure).

As soon as one attribute (i.e. time, space and form) is modified, a change during the assembly design occurs. Humans used to describe everything in the spatial dimension (e.g. one hour is measured with the little hand doing a complete round and coming back at its initial spatial position). That implies that time can be considered with spatial coordinates. This point is related to the concept of spatialized time or spatial time. As such, Bergson [Bergson, 1923] writes “I simultaneously get that I think in duration and that I am in duration”. In the context of AOD, this statement means that “I am designing the product assembly and I am aware of its assembly sequence in the AOD context”. The definition of spatial time from Heidegger [Heidegger, 1962] can be adapted to the moving process in assembly, which starts from the object initial position and finishes when the object reaches its final destination (and changes its spatial primitives). By adding temporal part, it enables the description of the object changes during product design stages.

As the purpose of this theory is to be adapted to every kind of design, it must remain a qualitative work. The object description - when (defined by assembly planner), where (by product architect), which form (by designer) - can be considered as complete with such kind of modeling technique (i.e. theory frame). Afterwards they all understand the transformation process (how, why) [Rodier et al., 2010]. When an object moves, properties from the past are preserved in the present (which is represented by the filiation relationship such as described later on). Therefore links between temporal regions exist. If the cause

Table 8: Spatial mereotopological primitives description

Name	Mereotopological description
<i>x is Part of y</i>	$x\mathbf{P}y := \forall z(z\mathbf{O}x \rightarrow z\mathbf{P}y)$
<i>x is InteriorPart of y</i>	$x\mathbf{I}P y := x\mathbf{P}y \wedge \neg x\mathbf{T}y$
<i>x Overlaps y</i>	$x\mathbf{O}y := \exists z(z\mathbf{P}x \wedge z\mathbf{P}y)$
<i>x is Discrete from y</i>	$x\mathbf{D}y := \neg x\mathbf{O}y$
<i>x Crosses y</i>	$x\mathbf{X}y := \neg x\mathbf{P}y \wedge \neg x\mathbf{D}y$
<i>x Tangent y</i>	$x\mathbf{T}y := \exists z(z\mathbf{P}x \wedge z\mathbf{B}y)$
<i>x is Boundary of y</i>	$x\mathbf{B}y := \forall z(z\mathbf{P}x \rightarrow z\mathbf{S}ty)$
<i>x Straddles y</i>	$x\mathbf{S}ty := \forall z(x\mathbf{I}P z \rightarrow z\mathbf{X}y)$
<i>Sum of <math>\phi</math> ers</i>	$\sigma x(\phi x) := \iota y(\forall w(w\mathbf{O}y \equiv \exists v(\phi v \wedge w\mathbf{O}v)))$

of the past changes is known, then the form of the object can be explained (but not predicted) in the present. As such, the object adaptation is regarded as understandable (e.g. why a planar pair has been used?). Indeed, product life is a series of adaptations and decisions that the designer has to make. As an object is related to numerous other objects, decisions impact the entire assembly (and not only the object).

### 3.3 Description of the spatial dimension of JANUS

The spatial dimension of the theory describes the spatial mereotopological relationships between spatial objects (e.g. mechanical parts) at a specific instant (e.g. at a very brief moment where a part touches another part) of the assembly process. Each spatial object is considered as a spatial region at a specific instant. A spatial region is defined as a portion of space occupied by an entity. Here a discrete space is considered as the description of the spatial position of an object according to others depends on their spatial mereotopological primitives. An empty region ( $\emptyset$ ), where no object is located, is also considered as a spatial region. The overall assembly ( $A$ ) is the region considering all regions forming the final product.

Spatial relationships capture the way how objects are related at a certain time [Del Mondo et al., 2010]. So the relative position of objects compared to others can be fully described [Rodier et al., 2010]. Each region is in relation with at least another region. If there is no contact between two objects, then the *Discrete D* primitive is used. From that moment on the relationship is valid for every spatial primitive (except discrete primitive), the letter **R**, which means *Relationship* is used. Consider that at time  $t$  there are two different parts:  $x$  and  $y$  (cf. Equation (3.1)).

$$\forall t \forall x \exists y / x_t \mathbf{R}y_t \quad (3.1)$$

Here the spatial primitives are introduced to formally describe part-to-part relationships. They are based on Smith's mereotopological primitives [Smith, 1996], and have already been addressed in Demoly et al. [Demoly et al., 2012b] and Kim et al. [Kim

et al., 2008] who have both focused their research efforts in assembly design and sequence planning. As for Salustri [Salustri, 2002], similar primitives – same properties but not the same names (e.g. *InteriorPart* **IP** is equivalent to *NonTangentialProperPart* **NTPP** or *TangentialProperPart* **TPP**) – have been introduced. Besides, the same author did not use the *Crosse* **X** primitive, which seems to be relevant in the above mentioned context and which is intensively used during assembly sequence planning.

By using mereotopological operators, eight spatial primitives have been represented in Figure 5 and described in Table 8. Here  $z$  is considered as a spatial region. In addition, “not  $x$ ” is a logical negation and an operation that can be interpreted as being true when its operand is false and false when its operand is true.

At the beginning of the design phase, part-to-part relationships are critical input for AOD. This information is then described into geometric skeletons in order to control assembly models in a central manner [Demoly et al., 2011a]. At a more detailed level, the condition  $\phi$  is used and states that a condition  $\phi$  in a single variable  $x$  is satisfied if and only if the sentence  $\phi x$  is true for at least one value of  $x$ . Based on this assumption, it is possible to introduce and define two entities (Figure 28 and Figure 29), on which the object form depends [Demoly et al., 2012b]:

- *Assembly skeleton*, which ensures assembly positioning, is represented by  $k$  (e.g. straight line representing the rotation and/or translation axis):  $k := \sigma z(\phi z) \rightarrow \forall z(\phi z \rightarrow (z\mathbf{T}x \wedge z\mathbf{T}y) \wedge (z\mathbf{P}x \wedge z\mathbf{P}y))$ ;
- *Interface skeleton*, which describes geometric boundaries (e.g. circle, square) used to build a functional surface and supported by an assembly skeleton, is represented by  $f$ :  $f := \sigma z(\phi z) \rightarrow \forall z(\phi z \rightarrow z\mathbf{B}x \wedge z\mathbf{B}y)$ .

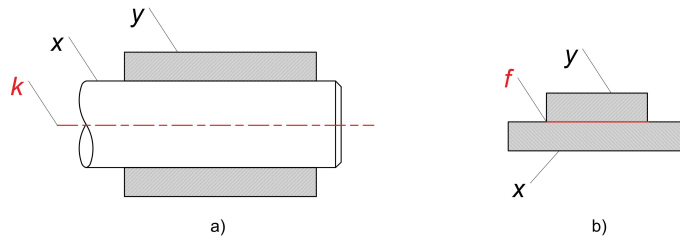


Figure 28: Geometric representation of  $k$  and  $f$  entities

Multiple assembly or interface skeletons on the same assembly, are respectively denoted  $k_1$  and  $k_2$  or  $f_1$  and  $f_2$ . Moreover, the interface skeleton type depends on the shape of the parts in physical contact. In Figure 29, parts can translate and rotate along the line 1-2 (assembly skeleton) and the disk enables knowing the edge in contact between both parts.

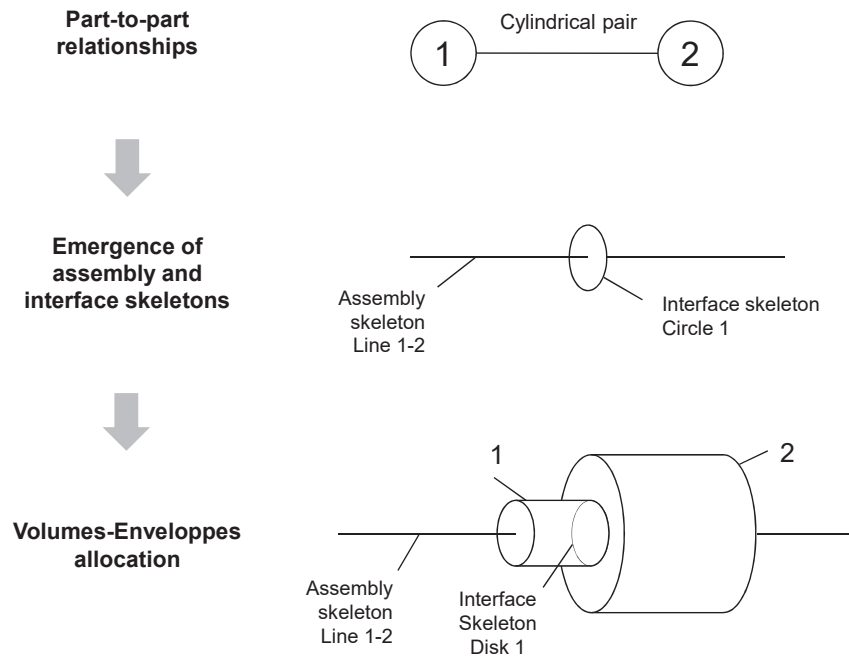


Figure 29: Example of cylindrical pair assembly and related skeletons (i.e. assembly and interface skeletons)

### 3.4 Description of the temporal description of JANUS

Based on the spatial part of JANUS, the temporal aspect with temporal relationships and temporal objects is then defined. The lifetime of an object is broken down into several smaller temporal objects [Hadjieleftheriou et al., 2002], which are created each time a part change occurs [Rodier et al., 2010]. The object change is generally associated to an event or a process. The purpose is to describe changes that affect objects during design and assembly process. As soon as an event occurs, changes occur at time  $t$ . An event indicates a discontinuity in a stable referential. There is a break between the “before” and the “after” of the event. In that sense, Sider [Sider, 2001] provides the following definition:  $x$  is an *Instantaneous Temporal Part of  $y$*  if, at instant  $t$ ,  $x$  exists at, but only at,  $t$ . Therefore the region is only spatial and there is no swept volume (i.e. no spatiotemporal region). On the other hand swept volumes – which are generated during assembly process – are used to understand and represent all product evolution over time. So process expresses a change during the intern evolution of the object and is composed of an initial event, a duration between both events and a final one.

Here each temporal object is considered as a **Temporal Region (TR)**. To understand the notion of temporal regions, mechanical assembly design history has to be considered as a story. It describes when the base part (first part to be assembled) is placed, how the second is assembled to the previous one and so on. As commonly admitted, this story

has temporal regions created at each new assembled spatial region. Temporal regions can be either intervals or instants [Sider, 2001]. Assembly operations are a kind of temporal regions. Thus, two types of temporality exists such as the temporality of the object itself (how it evolves over time) and the temporality of the object regarding to the others (which is shown with spatial primitives).

The relative chronology between assembly operations can be understood by using temporal relationships [Rodier et al., 2010]. Table 9 shows the temporal relationships between two intervals (third column), one interval and one point (fourth column), and two instants (with **Bt** the temporal boundary primitive) (fifth column). The proposed description is similar to Allen’s intervals [Allen, 1983] who formalizes topological relationships between temporal intervals. Here,  $a$ ,  $b$  and  $c$  are used to describe three temporal regions. Since the dimensions of the regions and primitives in the section 3.4 (i.e. temporal) are different from the one in section 3.3 (i.e. spatial), the authors have decided to use other letters for variables in order to remove any ambiguity. In the second line, the *TemporallyPart of* primitive does not enable knowing if  $b$  is located at the beginning or at the end of the interval  $a$ . If the assembly planner has a precised view of how the product will be assembled, then he can specify that  $b$  is *StartTemporallyPart of* or *FinishTemporallyPart of*  $a$ .

Temporal primitives are inspired from spatial mereotopological primitives and adapted to represent every temporal phenomenon that may be identified during an assembly process. Phenomena have been classified according to the type of mechanical assemblies. Just two of them can describe all types of assembly described in Table 10: *TemporallyPrecede*, denoted  $<$ , and *TemporallyTangent*, denoted **Tt**. The different types of assembly are considered as follows [Demoly et al., 2011b]: interconnected serial, serial, constrained serial and parallel. Both “interconnected serial” and “constrained” are constraint dependency relationships. The only difference is that in the latter, some part can be temporally constrained by other parts without being in contact with them (i.e. precedence constraint). For instance, in the “constrained” assembly type, *Part 2* must be assembled before *Part 3* and there is no contact between both parts. As a consequence, time is considered as linear [Renolen, 1999] and discrete.

## 3.5 Description of the spatiotemporal description of JANUS

### 3.5.1 Basic problems with spatiotemporal visualization

As the main interest is concerned about the description of assembly design evolution, a section dedicated to spatiotemporal dimension has been added. Each spatiotemporal object is considered as a spatiotemporal region such as swept volume during deformation, modification or transformation. The latter notion is the spatial region occupied by an object during a specific duration. By considering swept volume, the past coexists with the present. Skeletons do not have spatiotemporal parts, since spatiotemporal regions



Table 9: Temporal mereotopological primitives description and representation



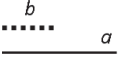

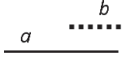

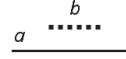
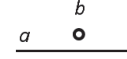
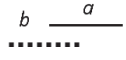
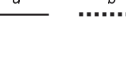
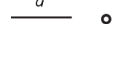





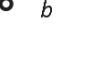
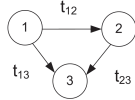
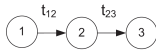
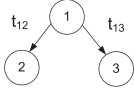
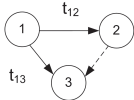
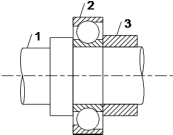
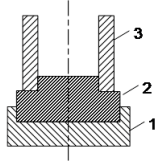
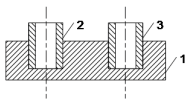
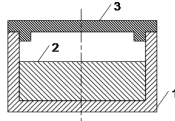
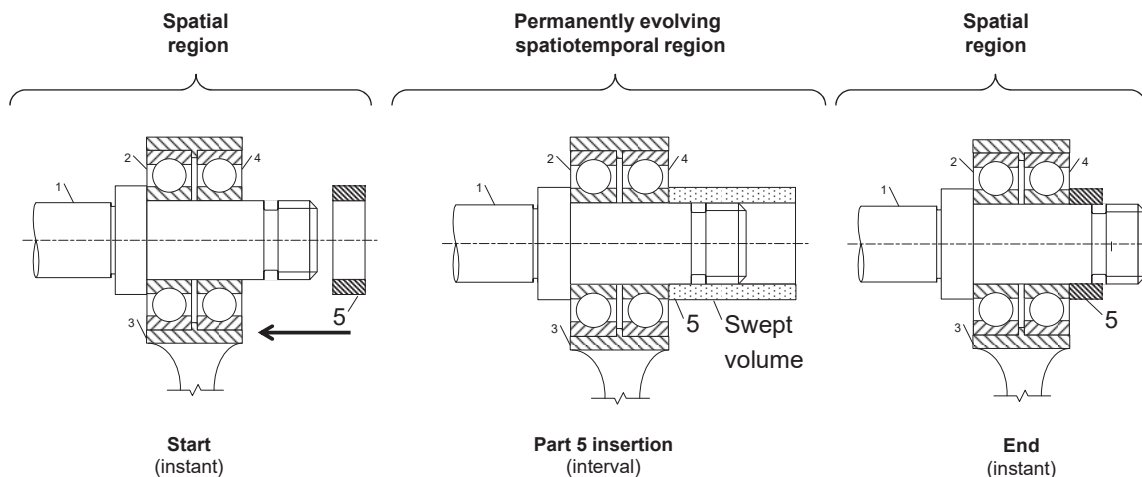
Name	Mereotopological description	Repres. with two intervals	Repres. with one interval and one instant	Repres. with two instants
<i>b</i> is TemporallyPart of <i>a</i>	$b\mathbf{P}ta := \forall c(c\mathbf{O}tb \rightarrow c\mathbf{P}ta)$			
<i>b</i> is StartTemporallyPart of <i>a</i>	$b\mathbf{P}tsa := \exists c(b\mathbf{P}ta \wedge c\mathbf{T}tb \wedge c < a)$			
<i>b</i> is FinishTemporallyPart of <i>a</i>	$b\mathbf{P}tfa := \exists c(b\mathbf{P}ta \wedge b\mathbf{T}tc \wedge a < c)$			
<i>b</i> is TemporallyInteriorPart of <i>a</i>	$b\mathbf{I}Pta := b\mathbf{P}ta \wedge \neg b\mathbf{P}tsa \vee \neg b\mathbf{P}tfa$			
<i>b</i> TemporallyOverlaps <i>a</i>	$b\mathbf{O}ta := \exists c(c\mathbf{P}tb \wedge c\mathbf{P}ta)$			
<i>a</i> TemporallyPrecedes <i>b</i>	$a < b := \exists c(a\mathbf{T}tc \wedge b\mathbf{P}tfc)$			
<i>b</i> is TemporallyTangent of <i>a</i>	$b\mathbf{T}ta := \exists c(c\mathbf{P}tsa \wedge c\mathbf{B}tb)$			
<i>b</i> is TemporallyEqual to <i>a</i>	$b =_t a := b\mathbf{I}Pta \wedge a\mathbf{I}Ptb$			



Table 10: Assembly types of three components

Assembly type	Interconnected serial	Serial	Parallel	Constrained
Directed graph				
Example				

consist of spatial and temporal components [Bittner, 2001]. For instance it is possible to see, during a change (e.g. Move of part 5), the path of an object and the related needed space to reach its final position (Figure 30).

Figure 30: Decomposition of *Part 5* insertion [Gruhier et al., 2015b]

Several problems concerning spatiotemporal visualization can be encountered by human. Actually human can only see the initial spatial region and the final spatial region during a move. He cannot percept the swept volume (spatiotemporal region). For instance during an assembly sequence, when a part is moving in order to cross the shaft, the designer cannot see the part being moved. He has to break down the move into several instants. This is due to human intelligence which is only able to see motionless object and which fixes the time. Humans cannot think in terms of continuous evolution. The spatiotemporal dimension of the theory has been defined like a sequence of pictures.

If an object must undergo two consecutive changes, then the final spatial region of the first change is the first spatial region of the second change. This type of description does not provide all information about what happens during intervals. A quantitative study has to be carried out with laws to know what happens in the interior of an interval.

### 3.5.2 Filiation relationships

Descartes [Descartes, 1996] made an experiment with wax. When he saw it, the wax was solid. But when he put it near fire, it melts. His conclusion is that with human understanding, the wax is wax and will always be wax whatever happens to it (e.g. transformation). Moreover Sider [Sider, 2001] and Hawley [Hawley, 2004] describes the object as persisting over time. When an object changes, its properties change (but not itself). The object will have attributes showing for instance if the object is deformed or not. For few changes (e.g. union or split), a new object is created from the “initial” one.

The same identity basis is shown with filiation relationships. It is a dependence relationship where the kinship is transmitted from the parent to the child. In assembly design, the parent can be an assembly and the child, which is based on its parent, can be a part. It enables the identification of every object and its uniqueness. Based on the identity concept, it is possible to distinguish an object from others. Two types of filiation relationships can be found during AOD phase [Del Mondo et al., 2010]:

- $\gamma$ : Continuation relationship (i.e. exactly the same object but at two different times);
- $\delta$ : Derivation relationship (i.e. a part of the identity from the original object is present in the changed object).

In the JANUS theory, the continuation relationship is considered as implicit. The latter always occurs between two objects with the same name at different instants. Derivation relationships occur when two objects are in relation with spatiotemporal primitives and do not have the same name.

### 3.5.3 Spatiotemporal primitives classification

Changes have spatiotemporal aspects and need spatiotemporal relationships to link spatial objects over time. The latter, which are used to link spatial and temporal dimensions [Del Mondo et al., 2010], can be described as follow:

- Between two spatial regions at two different instants (e.g. *Move*);
- From a spatial to an empty spatial region at two different instants (e.g. *Deletion*);
- From an empty spatial to a spatial region at two different instants (e.g. *Addition*).

Table 11: Part-to-part relationships definition for spatiotemporal primitives [Demoly et al., 2012b]

Relationship type	Definition
Contact	Physical contact relation between two components
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

As a consequence, spatiotemporal primitives express the change over space and time [Haddad, 2009], which provides an effective mechanism to visualize and communicate design and assembly intents [McKinney et al., 1996]. In that case, the evolution of an object may be seen as a succession of events and processes. Consider two different times  $t_i$  and  $t_j$  and two different parts  $x$  and  $y$  (cf. Equation (3.2)):

$$\forall t_i \neq t_j \forall x \exists y / x_{t_i} \mathbf{R} y_{t_j} \quad (3.2)$$

In rigid assembly, two parts during the assembly cannot physically *overlap*, on the contrary two spatiotemporal regions can *overlap* themselves at two different instants. For instance, several parts can be positioned on a shaft in a serial manner and so their spatiotemporal regions are overlapping but they are assembled at different times in order to have no physical overlap of their spatial regions. That is the reason why spatiotemporal primitives are needed to model the actual world. By describing new spatiotemporal primitives such as kinematic and technological pairs positioning (cf. Table 11), product design evolution is managed and understood. In Table 12, all spatiotemporal primitives are described and classified (based-on Haddad's [Haddad, 2009] classification) in terms of their changes over time, space and form.

For **Growing** and **Move**, the change occurs during the process: **SpatioTemporal Region (STR)** form changes, but also during the instant just after, i.e. change of **Spatial Region – Final (SRF)**. On the contrary for **Union**, changes in form just occur at the final spatial region. For instance, kinematic pairs undergo spatial and spatiotemporal changes, but part numbers stay identical. Moreover, the form of the part, which is moving, is not the same during the move (i.e. swept volume) and at the final stage of the evolution (i.e. part).

### 3.5.4 Kinematic pairs and Design changes description

Each spatiotemporal primitive is described in details using spatial and temporal mereotopological primitives (cf. Table 8 and Table 9). All kinematic pairs and design changes are listed on Figure 32. Table 13 and Table 14 show the mereotopological definitions of kinematic pairs positioning and design changes (such as **Addition**, **Growing** and so on).

Table 12: Classification of changes and associated spatiotemporal primitives

Changes types	Spatiotemporal primitives	Temporal change	Spatial change	Form change	Parts number change	Exemple of Event (E) or Process (P)
Movement	Kinematic pairs	•	•	STR+SRF	○	P: Place an object
	Rotation	•	•	STR+SRF	○	P: Orient an object
Replacement process	Permutation	•	•	STR+SRF	○	P: Designer oversight
Basic change	Addition	•	○	SRF	•	E: Missing object
	Deletion	•	○	SRF	•	E: Useless object
Restructuring Processes	Split	•	○	SRF	•	E: Complex assembly
	Union	•	○	SRF	•	E: Subassembly creation
Transformation	Growing	•	○	STR+SRF	○	P: Object too small
	Decrease	•	○	STR+SRF	○	P: Object too large
	ChangeOfForm	•	○	STR+SRF	○	P: Change of idea
	Deformation	•	○	STR+SRF	○	P: Rivet positioning
Technological pairs	Permanent with added part	•	•	STR+SRF	○	P: Place permanently an object
	Non-permanent with added part	•	•	STR+SRF	○	P: Place non-permanently an object
	Non-permanent without added part	•	•	STR+SRF	○	P: Place non-permanently an object

This kind of spatiotemporal primitives has been developed in order to enable the description of the product evolution over time in the context of AOD.

In table 13, the spatiotemporal primitive called **CylindricalOP** (here OP means operation) is described. The definition says that at the beginning *Part x* and *Part y* are **Discrete** and that *x is Overlapped by* its assembly skeleton *k*, which **is Discrete** with *y*. After that, *y moves* and *k Overlaps x* and *y*. As such, both *x* and *y* are aligned. At the end of the positioning, *y moves* again, **crosses x** and **are Tangent to** the interface skeleton *f*. Besides, JANUS theory does not consider the orientation during the part insertion. As such, **CylindricalOP** has the same mereotopological description than **PrismaticOP**.

The same primitive is also described in details in a specific graph (Figure 31) with its related legend (Table 15). The definition says that at the beginning *Part 1* and *Part 5* are **Discrete** and that *1 is Overlapped by* its assembly skeleton  $k_1$ , which **is Discrete** with *5*. After that, *5 moves* and  $k_1$  **Overlaps 1** and *5*. As such, both *1* and *5* are aligned. At the end of the positioning, *5 moves* again, **crosses 1** and **are Tangent to** the interface skeleton  $f_5$ . Processes are considered continuous, non-instantaneous (e.g. object cannot move from one point to another in one instant) and linear. **5 CylindricalOP 1** means that *1* is the base part and *5* is moving to be positioned in a cylindrical manner. For next descriptions,  $TRi$  always precedes  $TRi+1$ .

Moreover, each kinematic pair is represented in two dimensions in Table 16 and Table 17. *Part x* is static and *Part y* is moving to be positioned in a certain kinematic manner. The blue color represents *Part x*, the pink one *Part y* and the light pink one *Swpet volume y*. Skeletons are also part of the representation and guide the mereotopological description.

### 3.5.5 Technological pairs description

At a lower granularity, several technological pairs can be used when the product architect decides to design a rigid assembly: e.g. adhesive bonding, welding and so on. These relationships have been classified (cf. Table 18) and listed (cf. Figure 33) into four groups according to different criteria (e.g. added entities needed, and non-permanent assembly). Thus, it is not possible to have a non-permanent technological pair without added entities. Some rules have been set up, such as:

- *p* and *s* denote tools;
- *u* and *v* denote added entities;
- *x* and *y* denote both entities to be assembled.

According to the classification in Table 18, technological pairs positioning are described in Table 19, Table 20, Table 21, Table 22 and Table 23. Spatiotemporal descriptions expressed at  $TRi$  are not written anew at  $TRi+1$  in order to lighten the equations and to

Table 13: Mereotopological description of the “kinematic pairs positioning” primitives [Gruhier et al., 2015b]

Name	Mereotopological description	Representation	Comment
<b>Moves</b>	$(x\mathbf{T}k_1)_{TR1} \wedge (x\mathbf{T}k_2)_{TR2}$		$k_1$ and $k_2$ point, line or surface
<b>CylindricalOP</b>	$((x\mathbf{D}y) \wedge (x\mathbf{O}k) \wedge (y\mathbf{D}k))_{TR1} \wedge (y\mathbf{M}ove\ x)_{TR2} \wedge ((x\mathbf{D}y) \wedge (x\mathbf{O}k) \wedge (y\mathbf{O}k))_{TR3} \wedge (y\mathbf{M}ove\ x)_{TR4}$		$k$ line, $f$ surface
<b>PrismaticOP</b>	$\wedge((x\mathbf{X}y) \wedge (x\mathbf{O}k) \wedge (y\mathbf{O}k))_{TR5} \wedge (y\mathbf{M}ove\ x)_{TR6} \wedge ((x\mathbf{X}y) \wedge (x\mathbf{O}k) \wedge (y\mathbf{O}k) \wedge (y\mathbf{T}f))_{TR7}$		$k$ line, $f$ surface
<b>SphericalOP</b>	$(x\mathbf{D}y)_{TR1} \wedge (y\mathbf{M}ove\ x)_{TR2} \wedge ((x\mathbf{S}t\ y) \wedge (x\mathbf{I}P\ y) \wedge (x\mathbf{T}f) \wedge (y\mathbf{T}f) \wedge (x\mathbf{O}k) \wedge (y\mathbf{O}k))_{TR3}$		$k$ point, $f$ surface
<b>RevoluteOP</b>	$((x\mathbf{D}y) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{D}k_1))_{TR1} \wedge (y\mathbf{M}ove\ x)_{TR2} \wedge ((x\mathbf{D}y) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1))_{TR3} \wedge (y\mathbf{M}ove\ x)_{TR4} \wedge ((x\mathbf{X}y) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1))_{TR5} \wedge (y\mathbf{M}ove\ x)_{TR6} \wedge ((x\mathbf{T}y) \wedge (x\mathbf{X}y) \wedge (x\mathbf{O}k_1) \wedge (x\mathbf{T}k_2) \wedge (x\mathbf{T}f) \wedge (y\mathbf{O}k_1) \wedge (y\mathbf{T}k_2) \wedge (y\mathbf{T}f))_{TR7}$		$k_1$ line and $k_2$ plane, $f$ surface
<b>PlanarOP</b>	$((x\mathbf{D}y) \wedge (x\mathbf{T}k))_{TR1} \wedge (y\mathbf{M}ove\ x)_{TR2} \wedge ((x\mathbf{T}y) \wedge (x\mathbf{T}f) \wedge (y\mathbf{T}k) \wedge (f\mathbf{T}k))_{TR3}$		$k$ plane, $f$ surface
<b>PointContactOP</b>			$k$ point, $f$ point
<b>LineContactOP</b>			$k$ line, $f$ segment
<b>ScrewOP</b>	$((x\mathbf{D}y) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{D}k_1))_{TR1} \wedge (y\mathbf{M}ove\ x)_{TR2} \wedge ((x\mathbf{D}y) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1))_{TR3} \wedge (y\mathbf{M}ove\ x)_{TR4} \wedge ((x\mathbf{X}y) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1))_{TR5} \wedge (y\mathbf{M}ove\ x)_{TR6} \wedge ((x\mathbf{X}y) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1) \wedge (y\mathbf{T}f_1) \wedge (f_2\mathbf{O}f_3) \wedge (y\mathbf{T}f_2)) \wedge (x\mathbf{T}f_3))_{TR7}$		$k_1$ line, $f_1$ area, $f_2$ and $f_3$ elliptical curve
<b>Rotation</b>	$((x\mathbf{O}k_2) \wedge (k_1\mathbf{O}k_2) \wedge (k_1\mathbf{O}k_3) \wedge (k_2\mathbf{D}k_3))_{TR1} \wedge (x\mathbf{M}ove\ k_1)_{TR2} \wedge ((x\mathbf{O}k_2) \wedge (k_1\mathbf{O}k_2) \wedge (k_1\mathbf{O}k_3) \wedge (k_2\mathbf{O}k_3) \wedge (x\mathbf{O}k_3))_{TR3}$		$k_1$ point, $k_2$ and $k_3$ line

Table 14: Mereotopological description of the “design changes” primitives [Gruhier et al., 2015b]

Name	Mereotopological description	Representation	Comment
<b>Serial-Permutation</b>	$((xRz) \wedge (yRz) \wedge (xRy) \wedge (xTf) \wedge (xRk) \wedge (zTf_1) \wedge (zTf_2) \wedge (yTf_2) \wedge (yRk))_{TR1} \wedge (xMovez)_{TR2} \wedge ((xDz) \wedge (yRz) \wedge (xDy) \wedge (xRk) \wedge (zTf_2) \wedge (yTf_2) \wedge (yRk))_{TR3} \wedge (yMovez)_{TR4} \wedge ((xDz) \wedge (yDz) \wedge (xDy))_{TR5} \wedge (yMovez)_{TR6} \wedge ((xDz) \wedge (yRz) \wedge (xDy) \wedge (yTf_1) \wedge (yRk) \wedge (zTf_1))_{TR7} \wedge (xMovez)_{TR8} \wedge ((xRz) \wedge (yRz) \wedge (xRy) \wedge (xTf_2) \wedge (xRk) \wedge (zTf_1) \wedge (zTf_2) \wedge (yTf_1) \wedge (yRk))_{TR9}$		$k$ line, $f_1$ and $f_2$ surface
<b>Parallel-Permutation</b>	$((xRz) \wedge (yRz) \wedge (xDy) \wedge (xTf_1) \wedge (xRk_1) \wedge (zTf_1) \wedge (zTf_2) \wedge (yTf_2) \wedge (yRk_2) \wedge (f_1Df_2) \wedge (k_1Dk_2))_{TR1} \wedge ((xMovez) \wedge (yMovez))_{TR2} \wedge ((xDz) \wedge (yDz) \wedge (xDy))_{TR3} \wedge ((xMovez) \wedge (yMovez))_{TR4} \wedge ((xRz) \wedge (yRz) \wedge (xDy) \wedge (xTf_2) \wedge (xRk_2) \wedge (zTf_1) \wedge (zTf_2) \wedge (yTf_1) \wedge (yRk_1) \wedge (f_1Df_2) \wedge (k_1Dk_2))_{TR5}$		$k_1$ and $k_2$ line, $f_1$ and $f_2$ surface
<b>Addition</b>	$(xP\emptyset)_{TR1} \wedge ((xPA) \wedge (xTf))_{TR2}$	$\emptyset \rightarrow x \begin{array}{ c } \hline f \\ \hline \end{array}$	$f$ surface
<b>Deletion</b>	$((xPA) \wedge (xTf))_{TR1} \wedge (xP\emptyset)_{TR2}$	$x \begin{array}{ c } \hline f \\ \hline \end{array} \rightarrow \emptyset$	$f$ surface
<b>Split</b>	$((\emptyset Additiony) \wedge (\emptyset Additionz) \wedge (x\delta y) \wedge (x\delta z) \wedge (xTf_1))_{TR1} \wedge ((yPx) \wedge (zPx) \wedge (zTf_1) \wedge (yTf_2))_{TR2} \wedge ((xDeletion\emptyset) \wedge (zTf_1) \wedge (yTf_2))_{TR3}$		$f_1$ and $f_2$ surface
<b>Union</b>	$((xAddition\emptyset) \wedge (zTf_1) \wedge (yTf_2))_{TR1} \wedge ((yPx) \wedge (zPx) \wedge (zTf_1) \wedge (yTf_2))_{TR2} \wedge ((\emptyset Deletiony) \wedge (\emptyset Deletionz) \wedge (x\delta y) \wedge (x\delta z) \wedge (xTf_1))_{TR3}$		$f_1$ and $f_2$ surface
<b>ChangeOfForm</b>			
<b>Deformation</b>			
<b>Growing</b>	$(xTf_1)_{TR1} \wedge (xTf_2)_{TR2}$		$f_1$ and $f_2$ surface
<b>Decrease</b>			

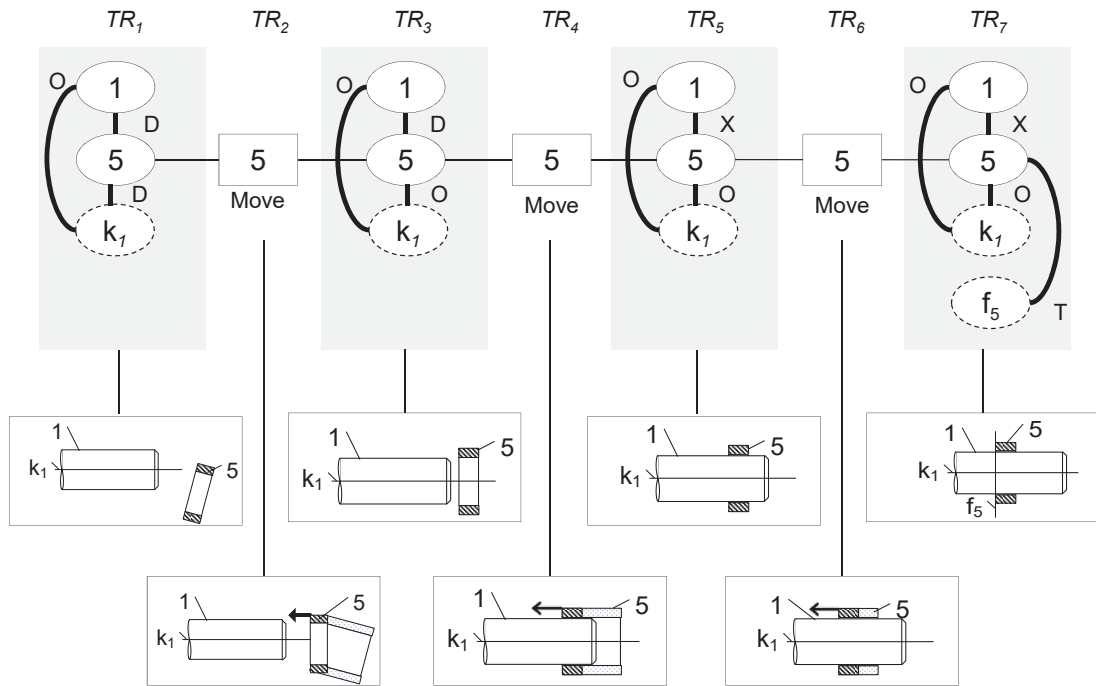


Figure 31: Description of *CylindricalOP* in a spatiotemporal graph [Gruhler et al., 2015b]

be more clear. Here, the authors have considered Kim et al.'s propositions related to the technological relationships description [Kim et al., 2008]. The goal of these descriptions is to trace the objects being assembled and also their tools, that is the reason why spatial and temporal primitives are needed.



Table 15: Legend of the spatiotemporal graph

Representation	Meaning
$\tau$	Spatial relationship (e.g. <b>Tangent</b> )
$\dashrightarrow$	Temporal relationship (e.g. <b>Precedence</b> )
Move	Spatiotemporal relationship (e.g. <b>Move</b> )
$TR_i$	Temporal region number $i$
5	Spatial region (e.g. named <i>Part 5</i> )
5	Spatiotemporal region (e.g. named <i>Swept Volume 5</i> )
k	Assembly skeleton (e.g. named $k$ )
f	Interface skeleton (e.g. named $f$ )

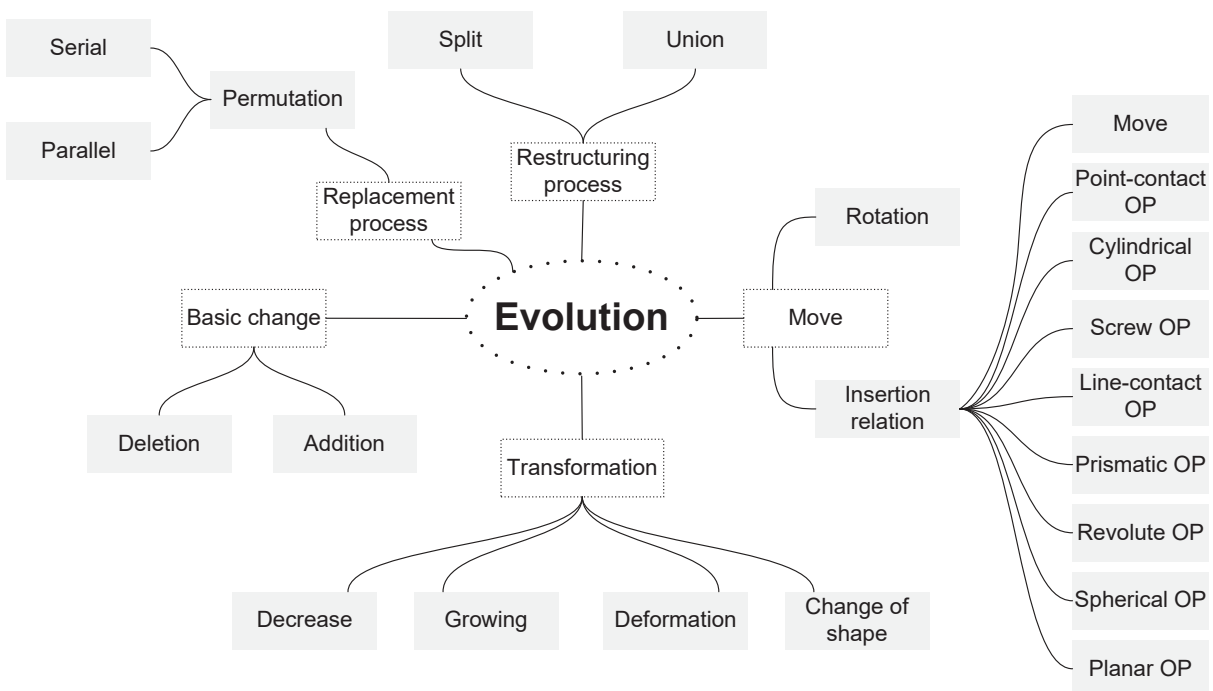


Figure 32: Cartography of the evolutions to be considered in PRONOIA2 [Gruhler et al., 2015c]

Table 16: Representation of the spatiotemporal kinematic pairs (1)

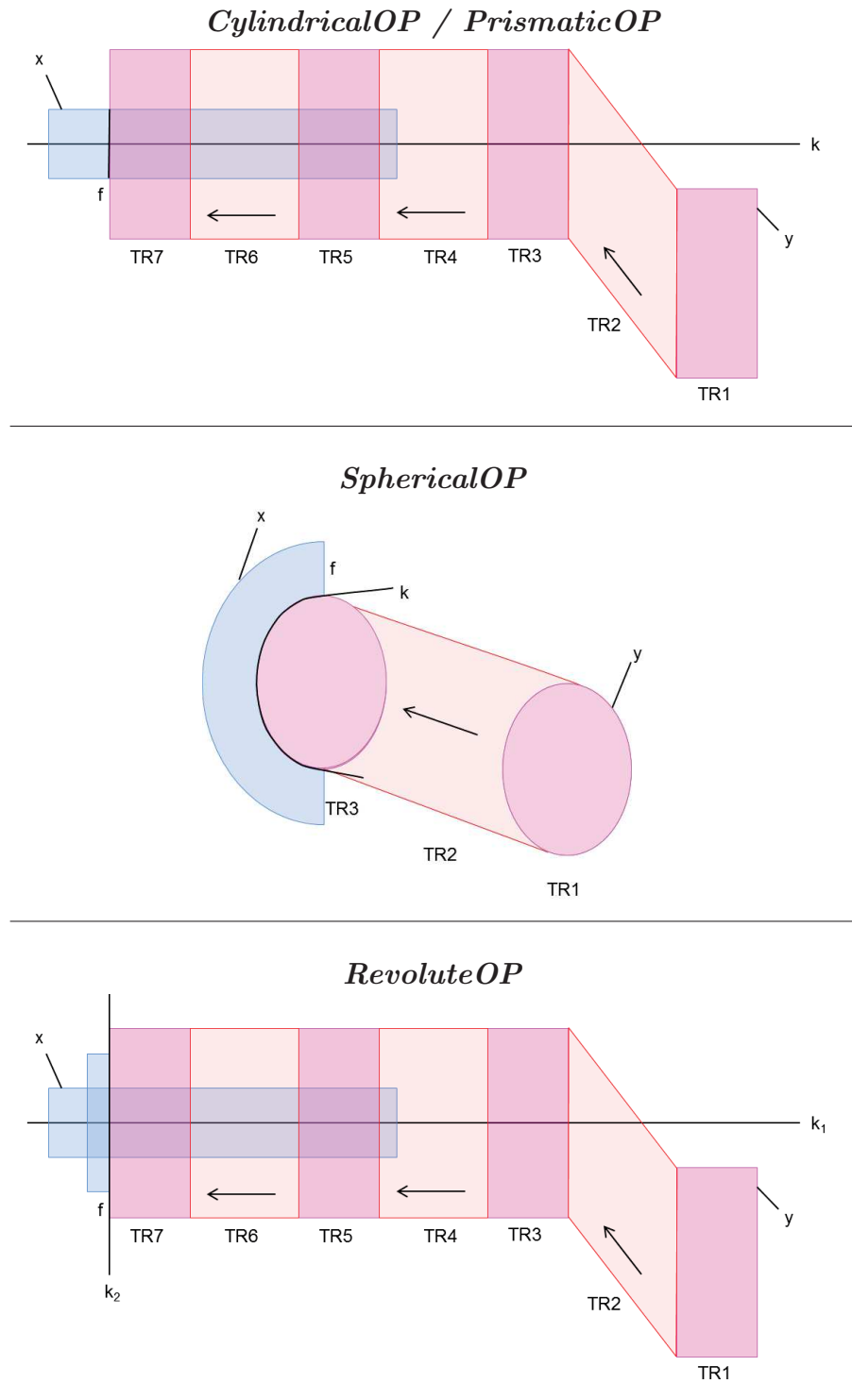


Table 17: Representation of the spatiotemporal kinematic pairs (2)  
*PlanarOP / PointContactOP / LineContactOP*

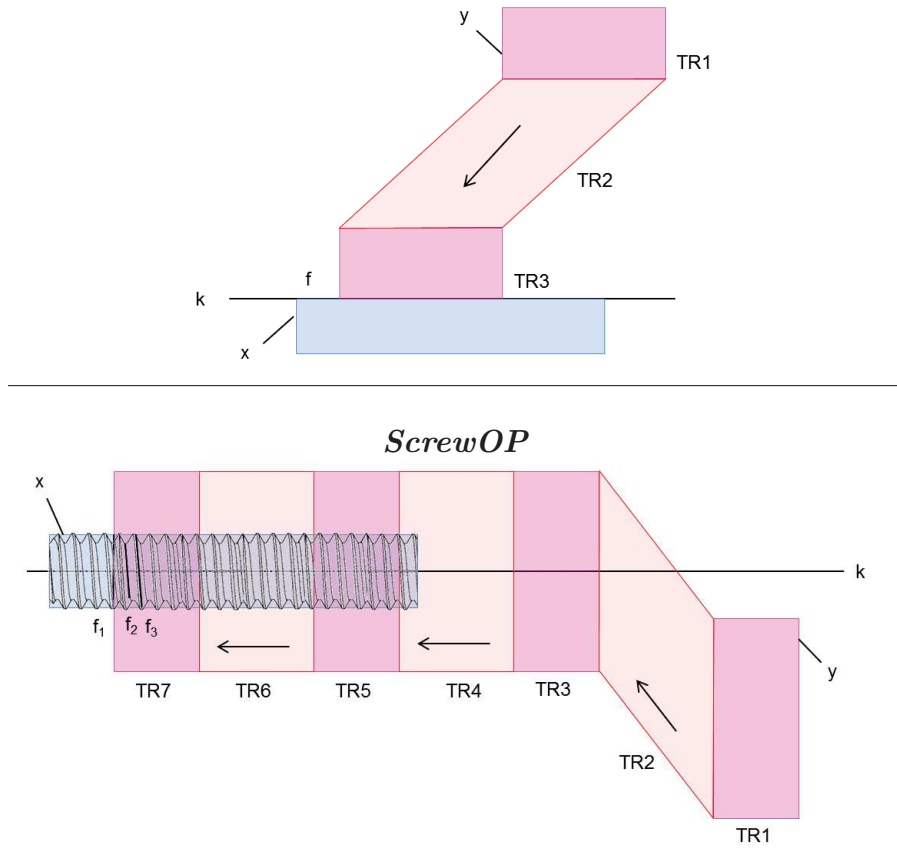
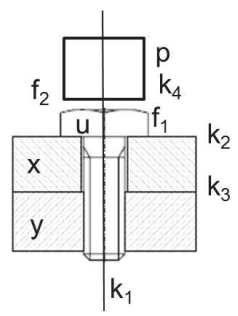


Table 18: Classification of technological assembly operations

Assembly Type		With added part	Without added part
Non-permanent		Mechanical Fastening	None
Permanent	Parts destruction	Gas Welding Brazing Adhesive Bonding	Arc Welding Spot Welding Clinching Crimping Interference Fit
	Fastening destruction	Nail Assembly Simple Rivet Double Rivet Blind Rivet Snap In Metal Stitching	None

Table 19: Mereotopological description of non-permanent assembly with added entities [Gruhler et al., 2015b]

Name	Mereotopological description	Representation	Comment
<b>Mechanical-Fastening-ByThreaded-Fasteners</b>	$((x\mathbf{T}y) \wedge (u\mathbf{D}x) \wedge (p\mathbf{T}u) \wedge (p\mathbf{T}f_2) \wedge$		$(k_1 \text{ line, } k_2, k_3 \text{ and } k_4 \text{ plane, } f_1 \text{ and } f_2 \text{ surface})$
	$(u\mathbf{T}f_2) \wedge (p\mathbf{T}k_4) \wedge (u\mathbf{T}k_4) \wedge (x\mathbf{T}k_3) \wedge$		
	$(y\mathbf{T}k_3) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1) \wedge (u\mathbf{O}k_1) \wedge$		
	$(p\mathbf{O}k_1))_{TR1} \wedge ((p\mathbf{CylindricalOP}x) \wedge$		
	$(u\mathbf{ScrewOP}x))_{TR2} \wedge (u\mathbf{X}x)_{TR3} \wedge$		
	$((p\mathbf{CylindricalOP}x) \wedge$		
	$(u\mathbf{ScrewOP}x))_{TR4} \wedge (u\mathbf{X}y)_{TR5} \wedge$		
	$((p\mathbf{CylindricalOP}x) \wedge$		
	$(u\mathbf{ScrewOP}x))_{TR6} \wedge ((u\mathbf{T}x) \wedge$		
$(u\mathbf{T}f_1) \wedge (x\mathbf{T}f_1) \wedge (u\mathbf{T}k_2) \wedge$			
$(x\mathbf{T}k_2))_{TR7} \wedge (p\mathbf{PrismaticOP}x)_{TR8} \wedge$			
$(p\mathbf{Deletion}\emptyset)_{TR9}$			

Four theorems are proposed below. For instance, with the mereotopological theory, theorem 1 describes the final stage of an assembly, which has undergone a heating process. It says that when part  $x$  is **deformed** by tool  $p$  or when  $x$  is **growing** with tool  $p$  and then parts  $x$  and  $y$  are **straddled**, then the assembly undergoes a heating process.

**Theorem 1.** Any assembly can be defined as assembly with heating process undergone, if and only if:

$$A := \exists x, y, p((x\mathbf{Deformation}p) \vee (x\mathbf{Growing}p))_{TRi} \wedge (x\mathbf{St}y)_{TRi+1}$$

**Theorem 2.** Any assembly can be defined as non-permanent assembly if and only if:

$$A := \exists x, y, p\neg(x\mathbf{Deformation}p)_{TRi} \wedge \neg(x\mathbf{St}y)_{TRi+1}$$

**Theorem 3.** Any assembly can be defined as permanent assembly without added entities if and only if:

$$A := \exists x, y, k((x\mathbf{St}y) \wedge (x\mathbf{St}k) \wedge (y\mathbf{St}k))_{TRi}$$

**Theorem 4.** Any assembly can be defined as permanent assembly with added entities if and only if:

$$A := \exists x, y, u(((u\mathbf{X}x) \wedge (u\mathbf{X}y) \wedge (u\mathbf{T}x)) \vee ((x\mathbf{St}u) \wedge (y\mathbf{St}u)))_{TRi}$$

Table 20: Mereotopological description of permanent assembly with added entities (1) [Gruhier et al., 2015b]

Name	Mereotopological description	Representation	Comment
<b>GasWelding</b>	$((x\mathbf{T}y) \wedge (x\mathbf{T}k) \wedge (y\mathbf{T}k))_{TR1} \wedge ((x\mathbf{Deformation}p) \wedge (y\mathbf{Deformation}p) \wedge (u\mathbf{Deformation}p) \wedge (u\mathbf{Move}x) \wedge (p\mathbf{Move}x))_{TR2} \wedge ((x\mathbf{St}y) \wedge (u\mathbf{St}x) \wedge (u\mathbf{St}y) \wedge (u\mathbf{D}p) \wedge (x\mathbf{D}p) \wedge (u\mathbf{T}f) \wedge (x\mathbf{T}f) \wedge (y\mathbf{T}f) \wedge (y\mathbf{D}p))_{TR3} \wedge (p\mathbf{Deletion}\emptyset)_{TR4}$		<i>k</i> line and <i>f</i> segment
<b>Soldering</b>	$((x\mathbf{T}y) \wedge (x\mathbf{T}k) \wedge (y\mathbf{T}k))_{TR1} \wedge ((u\mathbf{Move}x) \wedge (u\mathbf{Deformation}p) \wedge (p\mathbf{Move}x))_{TR2} \wedge ((u\mathbf{St}x) \wedge (u\mathbf{St}y) \wedge (u\mathbf{D}p) \wedge (x\mathbf{D}p) \wedge (u\mathbf{T}f) \wedge (x\mathbf{T}f) \wedge (y\mathbf{T}f) \wedge (y\mathbf{D}p))_{TR3} \wedge (p\mathbf{Deletion}\emptyset)_{TR4}$		<i>k</i> line and <i>f</i> segment
<b>Brazing</b>	$((x\mathbf{T}u) \wedge (x\mathbf{T}k_1) \wedge (u\mathbf{T}k_1) \wedge (u\mathbf{T}f_1) \wedge (x\mathbf{T}f_1))_{TR1} \wedge (y\mathbf{PrismaticOP}x)_{TR2} \wedge ((y\mathbf{T}k_2) \wedge (y\mathbf{T}u) \wedge (u\mathbf{T}k_2) \wedge (u\mathbf{T}f_2) \wedge (y\mathbf{T}f_2))_{TR3} \wedge ((p\mathbf{PrismaticOP}y) \wedge (u\mathbf{Deformation}p))_{TR4} \wedge ((p\mathbf{T}y) \wedge (u\mathbf{St}x) \wedge (u\mathbf{St}y) \wedge (y\mathbf{T}k_3) \wedge (p\mathbf{T}k_3) \wedge (p\mathbf{T}f_3) \wedge (y\mathbf{T}f_3))_{TR5} \wedge (p\mathbf{PrismaticOP}y)_{TR6} \wedge (p\mathbf{Deletion}\emptyset)_{TR7}$		<i>k</i> <sub>1</sub> , <i>k</i> <sub>2</sub> and <i>k</i> <sub>3</sub> line, <i>f</i> <sub>1</sub> , <i>f</i> <sub>2</sub> and <i>f</i> <sub>3</sub> surface
<b>SnapIn</b>	$((x\mathbf{D}u) \wedge (u\mathbf{O}k_1) \wedge (x\mathbf{O}k_1))_{TR1} \wedge ((u\mathbf{PrismaticOP}x) \wedge (u\mathbf{Deformation}x))_{TR2} \wedge ((u\mathbf{T}x) \wedge (u\mathbf{X}x) \wedge (u\mathbf{T}k_3) \wedge (x\mathbf{T}k_3) \wedge (u\mathbf{T}k_2) \wedge (x\mathbf{T}k_2))_{TR3} \wedge (u\mathbf{PrismaticOP}x) \wedge (u\mathbf{Deformation}x))_{TR4} \wedge (u\mathbf{St}x)_{TR5}$		<i>k</i> <sub>1</sub> line, <i>k</i> <sub>2</sub> and <i>k</i> <sub>3</sub> plane

Table 21: Mereotopological description of permanent assembly with added entities (2) [Gruhler et al., 2015b]

Name	Mereotopological description	Representation	Comment
<b>Nail-Assembly</b>	$((x\mathbf{T}y) \wedge (p\mathbf{T}u) \wedge (u\mathbf{D}x) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1) \wedge (x\mathbf{T}k_2) \wedge (y\mathbf{T}k_2) \wedge (u\mathbf{O}k_1) \wedge (p\mathbf{O}k_1) \wedge (p\mathbf{T}k_4) \wedge (u\mathbf{T}k_4) \wedge (p\mathbf{T}f_2) \wedge (u\mathbf{T}f_2))_{TR1} \wedge ((p\mathbf{PrismaticOP}x) \wedge (u\mathbf{PrismaticOP}x) \wedge (x\mathbf{Deformation}u))_{TR2} \wedge (u\mathbf{X}x)_{TR3} \wedge ((p\mathbf{PrismaticOP}x) \wedge (u\mathbf{PrismaticOP}x) \wedge (x\mathbf{Deformation}u) \wedge (y\mathbf{Deformation}u))_{TR4} \wedge ((u\mathbf{X}x) \wedge (u\mathbf{X}y))_{TR5} \wedge ((p\mathbf{PrismaticOP}x) \wedge (u\mathbf{PrismaticOP}x) \wedge (x\mathbf{Deformation}u) \wedge (y\mathbf{Deformation}u))_{TR6} \wedge ((u\mathbf{X}x) \wedge (u\mathbf{X}y) \wedge (u\mathbf{T}x) \wedge (u\mathbf{T}k_3) \wedge (x\mathbf{T}k_3) \wedge (u\mathbf{T}f_1) \wedge (x\mathbf{T}f_1))_{TR7} \wedge (p\mathbf{PrismaticOP}x)_{TR8} \wedge (p\mathbf{D}u)_{TR9} \wedge (p\mathbf{PrismaticOP}x)_{TR10} \wedge (p\mathbf{Deletion}\emptyset)_{TR11}$		$k_1$ line, $k_2$ , $k_3$ and $k_4$ plane, $f_1$ and $f_2$ surface
<b>Metal-Stitching</b>	$((x\mathbf{T}y) \wedge (p\mathbf{X}x) \wedge (p\mathbf{X}y) \wedge (u\mathbf{I}p) \wedge (x\mathbf{T}k_2) \wedge (y\mathbf{T}k_2))_{TR1} \wedge ((p\mathbf{PrismaticOP}x) \wedge (u\mathbf{PrismaticOP}x))_{TR2} \wedge ((p\mathbf{X}x) \wedge (p\mathbf{X}y) \wedge (p\mathbf{T}x) \wedge (p\mathbf{T}y))_{TR3} \wedge ((u\mathbf{PrismaticOP}x) \wedge (x\mathbf{Deformation}u))_{TR4} \wedge (u\mathbf{X}x)_{TR5} \wedge ((u\mathbf{PrismaticOP}x) \wedge (y\mathbf{Deformation}u))_{TR6} \wedge ((u\mathbf{T}p) \wedge (u\mathbf{X}y))_{TR7} \wedge (u\mathbf{Deformation}x)_{TR8} \wedge ((u\mathbf{X}y) \wedge (u\mathbf{X}y) \wedge (u\mathbf{T}x) \wedge (p\mathbf{X}x) \wedge (p\mathbf{X}y) \wedge (u\mathbf{T}f) \wedge (x\mathbf{T}f) \wedge (x\mathbf{T}k_1) \wedge (u\mathbf{T}k_1))_{TR9} \wedge (u\mathbf{Deformation}p)_{TR10} \wedge ((u\mathbf{T}y) \wedge (u\mathbf{T}k_3) \wedge (y\mathbf{T}k_3))_{TR11} \wedge (p\mathbf{PrismaticOP}x)_{TR12} \wedge (p\mathbf{Deletion}\emptyset)_{TR13}$		$k_1$ , $k_2$ and $k_3$ plane and $f$ surface

Table 22: Mereotopological description of permanent assembly with added entities (3) [Gruhier et al., 2015b]

Name	Mereotopological description	Representation	Comment
<b>SimpleRivet</b>	$((u\mathbf{B}s) \wedge (\emptyset\mathbf{Addition}y) \wedge (u\mathbf{B}f_2) \wedge (u\mathbf{O}k_1) \wedge (s\mathbf{O}k_1))_{TR1} \wedge$ $(y\mathbf{PrismaticOP}s)_{TR2} \wedge ((u\mathbf{T}y) \wedge (u\mathbf{X}y) \wedge (\emptyset\mathbf{Addition}x) \wedge$ $(u\mathbf{T}k_2) \wedge (y\mathbf{T}k_2) \wedge (y\mathbf{O}k_1))_{TR3} \wedge (x\mathbf{PrismaticOP}y)_{TR4} \wedge ((u\mathbf{T}y) \wedge$ $(u\mathbf{X}x) \wedge (\emptyset\mathbf{Addition}p) \wedge (x\mathbf{O}k_1) \wedge (x\mathbf{T}k_3) \wedge (y\mathbf{T}k_3))_{TR5} \wedge$ $(p\mathbf{PrismaticOP}x)_{TR6} \wedge ((p\mathbf{B}u) \wedge (p\mathbf{O}k_1) \wedge (u\mathbf{B}f_1) \wedge (p\mathbf{B}f_1))_{TR7} \wedge$ $((p\mathbf{PrismaticOP}x) \wedge (u\mathbf{Deformation}p))_{TR8} \wedge ((u\mathbf{T}x) \wedge (p\mathbf{T}x) \wedge$ $(u\mathbf{T}k_4) \wedge (p\mathbf{T}k_4))_{TR9} \wedge (p\mathbf{PrismaticOP}x)_{TR10} \wedge (p\mathbf{Deletion}\emptyset)_{TR11}$		$k_1$ line, $k_2$ , $k_3$ and $k_4$ plane, $f_1$ and $f_2$ surface
<b>DoubleRivet</b>	$((v\mathbf{T}s) \wedge (\emptyset\mathbf{Addition}y) \wedge (v\mathbf{O}k_1) \wedge (v\mathbf{T}k_5) \wedge (s\mathbf{O}k_1))_{TR1} \wedge$ $(y\mathbf{PrismaticOP}s)_{TR2} \wedge ((\emptyset\mathbf{Addition}x) \wedge (v\mathbf{T}k_2) \wedge (y\mathbf{T}k_2) \wedge$ $(v\mathbf{X}y) \wedge (y\mathbf{O}k_1))_{TR3} \wedge (x\mathbf{PrismaticOP}y)_{TR4} \wedge ((\emptyset\mathbf{Addition}u) \wedge$ $(x\mathbf{O}k_1) \wedge (x\mathbf{T}k_3) \wedge (y\mathbf{T}k_3) \wedge (v\mathbf{X}x))_{TR5} \wedge (u\mathbf{PrismaticOP}x)_{TR6} \wedge$ $((\emptyset\mathbf{Addition}p) \wedge (u\mathbf{T}k_5) \wedge (u\mathbf{S}t v) \wedge (u\mathbf{X}x) \wedge (u\mathbf{X}y))_{TR7} \wedge$ $(p\mathbf{PrismaticOP}x)_{TR8} \wedge ((p\mathbf{B}u) \wedge (p\mathbf{O}k_1) \wedge (u\mathbf{B}f_1) \wedge (p\mathbf{B}f_1))_{TR9} \wedge$ $((p\mathbf{PrismaticOP}x) \wedge (u\mathbf{Deformation}p) \wedge$ $(v\mathbf{Deformation}s))_{TR10} \wedge ((u\mathbf{T}x) \wedge (p\mathbf{T}x) \wedge (u\mathbf{T}k_4) \wedge (p\mathbf{T}k_4) \wedge$ $(v\mathbf{B}f_2) \wedge (v\mathbf{T}y) \wedge (v\mathbf{B}s))_{TR11} \wedge (p\mathbf{PrismaticOP}x)_{TR12} \wedge$ $(p\mathbf{Deletion}\emptyset)_{TR13}$		$k_1$ line, $k_2$ , $k_3$ , $k_4$ and $k_5$ plane, $f_1$ and $f_2$ surface
<b>BlindRivet</b>	$((x\mathbf{T}y) \wedge (p\mathbf{X}v) \wedge (p\mathbf{T}u) \wedge (u\mathbf{D}x) \wedge (u\mathbf{X}v) \wedge (x\mathbf{T}k_2) \wedge$ $(y\mathbf{T}k_2) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1) \wedge (v\mathbf{O}k_1) \wedge (u\mathbf{O}k_1) \wedge (p\mathbf{O}k_1))_{TR1} \wedge$ $(p\mathbf{PrismaticOP}x)_{TR2} \wedge (u\mathbf{X}x)_{TR3} \wedge (p\mathbf{PrismaticOP}x)_{TR4} \wedge$ $(u\mathbf{X}y)_{TR5} \wedge (p\mathbf{PrismaticOP}x)_{TR6} \wedge ((u\mathbf{T}x) \wedge (u\mathbf{T}k_3) \wedge$ $(u\mathbf{T}f) \wedge (x\mathbf{T}f) \wedge (x\mathbf{T}k_3))_{TR7} \wedge ((v\mathbf{PrismaticOP}x) \wedge$ $(u\mathbf{Deformation}p))_{TR8} \wedge (u\mathbf{T}y)_{TR9} \wedge ((v\mathbf{Split}a) \wedge (v\mathbf{Split}b) \wedge$ $(p\mathbf{PrismaticOP}x))_{TR10} \wedge ((p\mathbf{D}x) \wedge (a\mathbf{IP}u) \wedge (b\mathbf{IP}p) \wedge$ $(p\mathbf{Deletion}\emptyset) \wedge (b\mathbf{Deletion}\emptyset))_{TR11}$		$k_1$ line, $k_2$ and $k_3$ plane, $f$ surface

Table 23: Mereotopological description of permanent assembly without added entities [Gruhier et al., 2015b]

Name	Mereotopological description	Representation	Comment
<b>ArcWelding</b>	$((x\mathbf{T}y) \wedge (p\mathbf{D}x) \wedge (p\mathbf{D}y) \wedge (x\mathbf{T}k) \wedge$ $(y\mathbf{T}k))_{TR1} \wedge ((x\mathbf{Deformation}p) \wedge$ $(y\mathbf{Deformation}p))_{TR2} \wedge ((x\mathbf{St}y) \wedge$ $(x\mathbf{St}k) \wedge y\mathbf{St}k) \wedge (p\mathbf{Deletion}\emptyset))_{TR3}$		$k$ plane
<b>SpotWelding</b>	$((x\mathbf{T}y) \wedge (p\mathbf{D}x) \wedge (s\mathbf{T}y) \wedge (p\mathbf{O}k_1) \wedge$ $(s\mathbf{O}k_1) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1) \wedge$ $(s\mathbf{T}f_2) \wedge (s\mathbf{T}k_2) \wedge (y\mathbf{T}k_2) \wedge$ $(y\mathbf{T}f_2) \wedge (x\mathbf{T}k_3) \wedge (y\mathbf{T}k_3))_{TR1} \wedge$ $(p\mathbf{PrismaticOP}x)_{TR2} \wedge$ $((p\mathbf{T}x) \wedge (p\mathbf{T}f_1) \wedge (x\mathbf{T}k_4) \wedge$ $(p\mathbf{T}k_4))_{TR3} \wedge ((x\mathbf{Deformation}p) \wedge$ $(y\mathbf{Deformation}s))_{TR4} \wedge ((x\mathbf{St}y) \wedge$ $(x\mathbf{St}k_3) \wedge (y\mathbf{St}k_3))_{TR5} \wedge$ $((p\mathbf{PrismaticOP}x) \wedge$ $(s\mathbf{PrismaticOP}x))_{TR6} \wedge$ $((p\mathbf{Deletion}\emptyset) \wedge (s\mathbf{Deletion}\emptyset))_{TR7}$		$k_1$ line, $k_2, k_3$ and $k_4$ plane, $f_1$ and $f_2$ surface
<b>Clinching</b>	$((x\mathbf{T}s) \wedge (x\mathbf{T}k_2) \wedge (x\mathbf{T}k_1) \wedge (x\mathbf{T}f_1) \wedge$ $(p\mathbf{T}k_2) \wedge (p\mathbf{T}f_2) \wedge (s\mathbf{T}k_1) \wedge (s\mathbf{T}f_1) \wedge$ $(x\mathbf{T}y) \wedge (y\mathbf{T}p) \wedge (y\mathbf{T}k_4) \wedge (y\mathbf{T}k_2) \wedge$ $(y\mathbf{T}f_2))_{TR1} \wedge ((p\mathbf{PrismaticOP}x) \wedge$ $(x\mathbf{Deformation}s) \wedge$ $(y\mathbf{Deformation}s))_{TR2} \wedge$ $((s\mathbf{B}x) \wedge (p\mathbf{B}y) \wedge (x\mathbf{St}y) \wedge$ $(x\mathbf{St}k_4) \wedge (y\mathbf{St}k_4))_{TR3} \wedge$ $(p\mathbf{PrismaticOP}x)_{TR4} \wedge$ $((p\mathbf{Deletion}\emptyset) \wedge (s\mathbf{Deletion}\emptyset))_{TR5}$		$k_3$ line, $k_1, k_2$ and $k_4$ plane, $f_1$ and $f_2$ surface
<b>Crimping</b>	$((x\mathbf{X}y) \wedge (x\mathbf{T}k_3) \wedge (y\mathbf{T}k_3) \wedge (p\mathbf{T}y) \wedge$ $(p\mathbf{X}y) \wedge (x\mathbf{O}k_1) \wedge (y\mathbf{O}k_1) \wedge (y\mathbf{T}k_2) \wedge$ $(p\mathbf{T}k_2))_{TR1} \wedge ((p\mathbf{Decrease}x) \wedge$ $(y\mathbf{Deformation}p) \wedge$ $(x\mathbf{Deformation}y))_{TR2} \wedge ((p\mathbf{B}y) \wedge$ $(y\mathbf{St}x) \wedge (x\mathbf{St}k_3) \wedge (y\mathbf{St}k_3))_{TR3} \wedge$ $(p\mathbf{PrismaticOP}x)_{TR4} \wedge$ $(p\mathbf{Deletion}\emptyset))_{TR5}$		$k_1$ line and $k_2, k_3$ plane
<b>Interference-Fit</b>	$((x\mathbf{D}y) \wedge (x\mathbf{O}k_1) \wedge$ $(y\mathbf{O}k))_{TR1} \wedge ((y\mathbf{Growing}x) \wedge$ $(y\mathbf{PrismaticOP}x))_{TR2} \wedge$ $(x\mathbf{X}y)_{TR3} \wedge (y\mathbf{Decrease}x)_{TR4} \wedge$ $((x\mathbf{St}k_2) \wedge (y\mathbf{St}k_2) \wedge (x\mathbf{St}y))_{TR5}$		$k_1$ line and $k_2$ plane



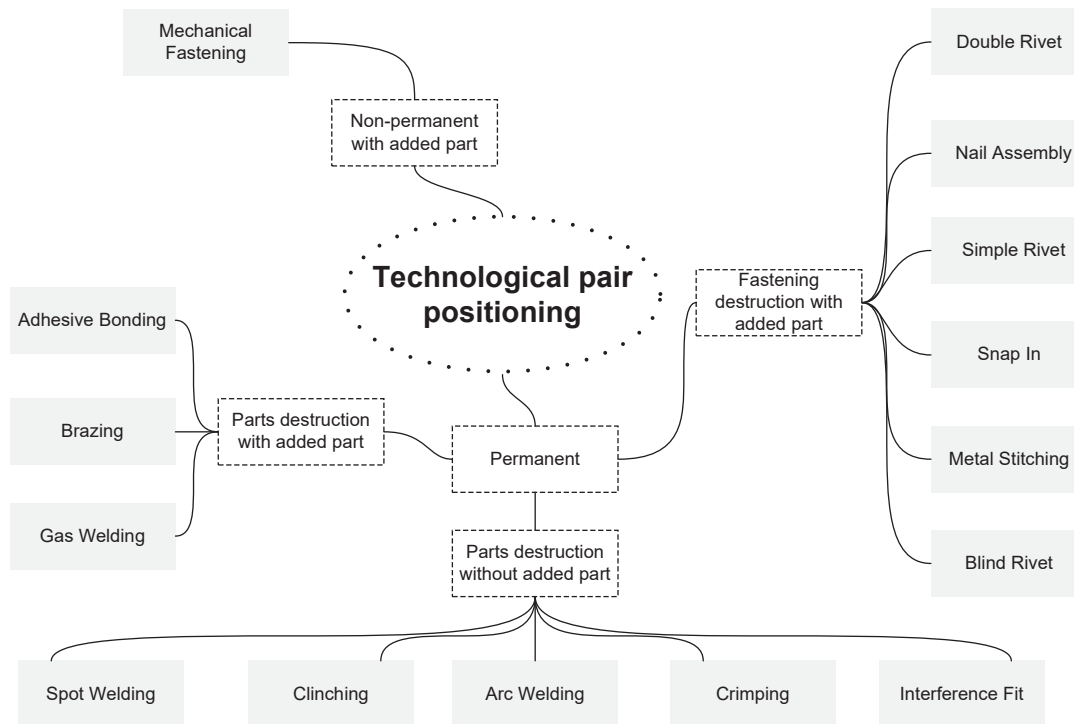


Figure 33: Cartography of the technological pairs to be considered in PRONOA2

### 3.6 Summary

Built on the previous mereotopological theories classification, a novel mereotopological theory in the context of AOD, called JANUS, covering three dimensions: spatial, temporal and spatiotemporal has been proposed and developed, so as to describe product-process relationships. Spatial and temporal objects and primitives have been defined, as well as the associations between both dimensions. Indeed new spatiotemporal objects and primitives have been introduced in order to describe product-process knowledge and information.

By considering the spatiotemporal dimension and the associated mereotopology-based theory, information has been added to technical objects so as to express their history in a consistent and understandable manner. This description aids to understand phenomena during AOD process and describes products with a philosophy of perdurantism. The actual stake is to get a long term dynamic vision of the space over time in order to facilitate the understanding of assembly and design changes. We conclude this chapter by the necessity of implementing this proposed mereotopological theory in an ontology so as to be machine interpretable.

# Chapter 4

## Development and implementation of the formal ontology – PRONOIA2

### Contents

---

4.1	Introduction . . . . .	78
4.2	Overview of the research approach . . . . .	78
4.3	Design rationale . . . . .	82
4.4	Meta-ontology description . . . . .	82
4.5	Assembly-ontology description . . . . .	84
4.6	OWL-DL restriction . . . . .	88
4.6.1	General information on OWL-DL . . . . .	88
4.6.2	Restriction in the meta- and assembly-ontology . . . . .	90
4.7	SWRL rules . . . . .	95
4.8	Summary . . . . .	97

---

## 4.1 Introduction

The implementation of JANUS theory is presented in this chapter. The first paragraph introduces the PRONOAIA2 model and its different steps in which the ontology is inserted. The second paragraph defines the three layers of the ontology, such as meta-, domain- and application-ontology (also described in Figure 34). Then, the third paragraph presents the implementation of the theory in Protégé Editor using OWL language. Then, the next paragraphs describe the restrictions (with DL) and rules (with SWRL) added to the model in order to reason on it and be able to check the information consistency. As such, the theory is completely formalized. The last paragraph summarizes the main results.

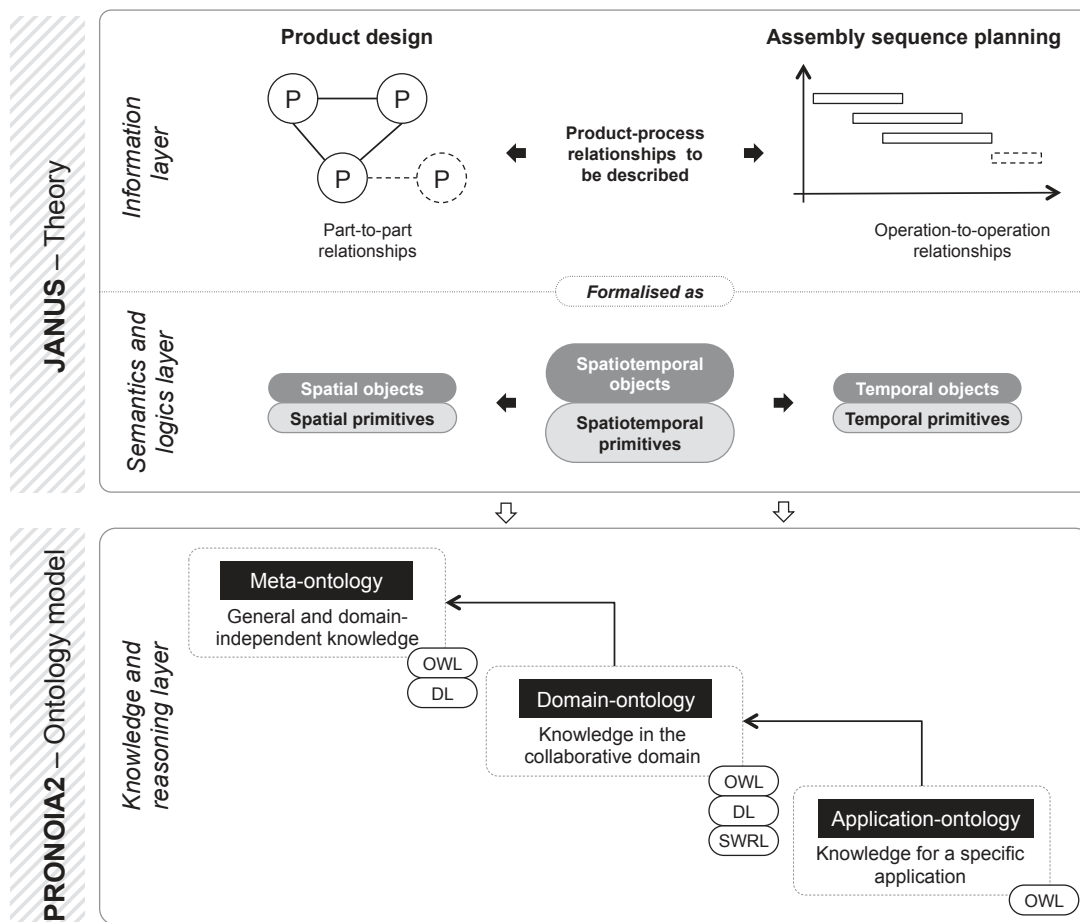


Figure 34: PRONOAIA2 ontology model built from JANUS theory

## 4.2 Overview of the research approach

The novel approach, called PRONOAIA2, is based on the PRONOAIA approach [Demoly et al., 2012b] and extends it in the temporal and spatiotemporal dimension in order to

have a more realistic model. The model aids the designer to check information consistency in the early design phases. Product information can be accessed, stored, served and reused throughout the entire product lifecycle [Zhong et al., 2013] by using a dedicated ontology. The ontology is implemented using Protégé, an ontology management tool [Sun et al., 2010] and OWL2 (extension of OWL).

The main strength of Protégé compared to other similar systems are: its user interface, the extendibility using plug-ins, the functionality that the plug-ins provide as well as the different formats that can be imported and exported [Chang, 2008]. As illustrated in Figure 34, the PRONOIA2 ontology is divided into three levels, such as the meta-ontology, the domain-ontology and the application-ontology.

Here meta-ontology gives the methodology, which justifies why this ontology is created and explains why it works that way. It is a generic and top-level ontology, which captures general, domain independent knowledge [Fensel et al., 2007]. A meta-ontology is composed of meta-classes. A meta-class is a class whose instances are themselves classes. It is a template for building classes, in the same way that classes are templates for building instances [Tudorache, 2006].

Domain-ontology captures knowledge in the collaborative domain that is shared with developers [Rahmani and Thomson, 2012] to provide a common understanding of the domain [Chang, 2008]. It is dedicated to one closed domain that realized specific tasks and adapted to one business (i.e. assembly, manufacturing process, etc.). It also gives the possibility of creating a knowledge base specific to the application domain [Zanni-Merk et al., 2011].

Finally the application-ontology, captures the knowledge necessary for a specific application [Fensel et al., 2007]. As such, ontology enables to externalize tacit knowledge. Tacit knowledge covers knowledge that is unarticulated and tied to the senses, movement skills, physical experiences and intuition. It is different from explicit knowledge, which is uttered and captured in drawings and writings. It is accessible through consciousness and rooted in action, procedures, routines, commitments, ideas, values and emotions [Nonaka and von Krogh, 2009].

Here the overview of the PRONOIA2 approach is presented (cf. Figure 35):

**Step 1** Based on an eBOM, the product architect draws the part-to-part relationships (i.e. contact, precedence, kinematic and technological) in a directed graph. As such, functional requirements are fulfilled.

**Step 2** Based on the part-to-part relationships, the assembly sequence is generated by the **ASDA** (**A**ssembly **S**equences **D**efinition **A**lgorithm) algorithm [Demoly et al., 2011b]. Possible sub-assemblies, as well as their types (i.e. serial, parallel, etc.) and the base parts are detected. This information can be illustrated in a Gantt chart (left side of Figure 35). Moreover spatial information on the directed graph are

converted into spatial mereotopological primitives (right side of Figure 35). Then, spatiotemporal primitives are known with previous information and are represented in a space-time graph.

**Step 3** Based on previous information, the mereotopological description is refined. As such, the product is structured and its relationships are defined at the early design stage.

**Step 4** The JANUS mereotopological theory is implemented in Protégé using 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, to make this information available for reuse in MOL/EOL and to check the information consistency. As shown in Figure 36, the ontology model is structured into three levels, such as the meta-ontology, the domain-ontology and the application-ontology.

Compared to the two dimensions studied in PRONOIA, the extended PRONOIA2 model integrates three dimensions, such as spatial, temporal and spatiotemporal. The theory, the ontology model and the rules link spatial, temporal and new spatiotemporal notions and allows to have a more complete and realistic model in order to qualitatively describe the product-process evolution.

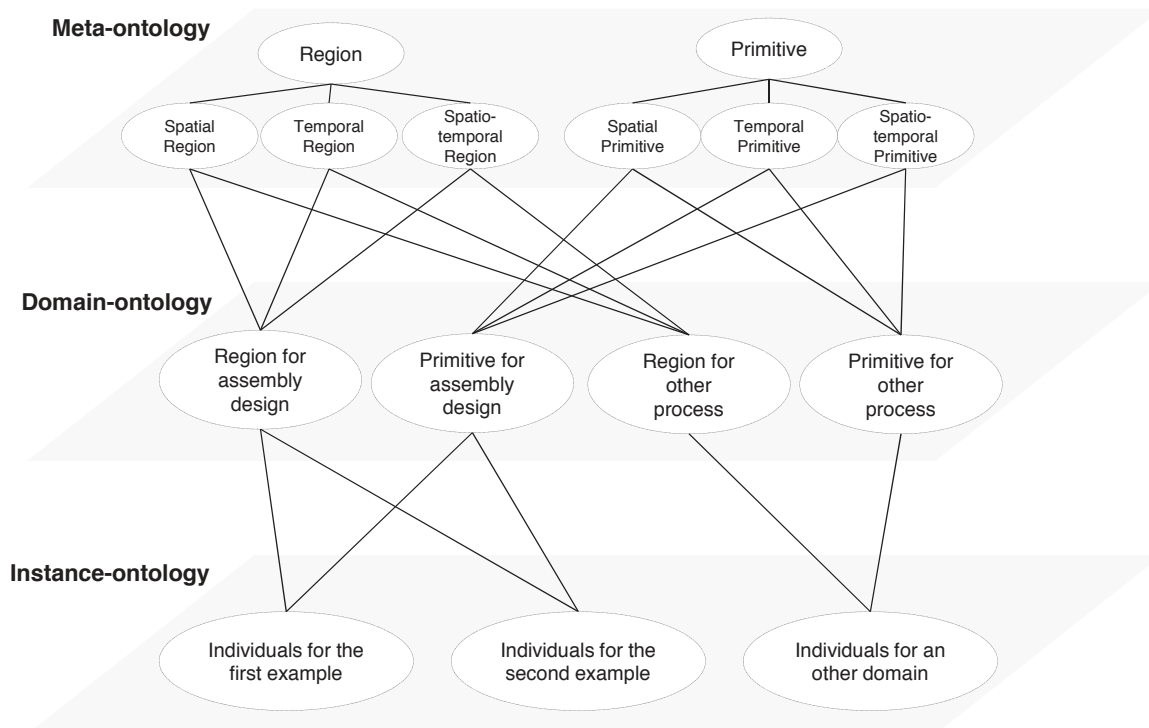


Figure 36: Proposed framework of the PRONOIA2 approach

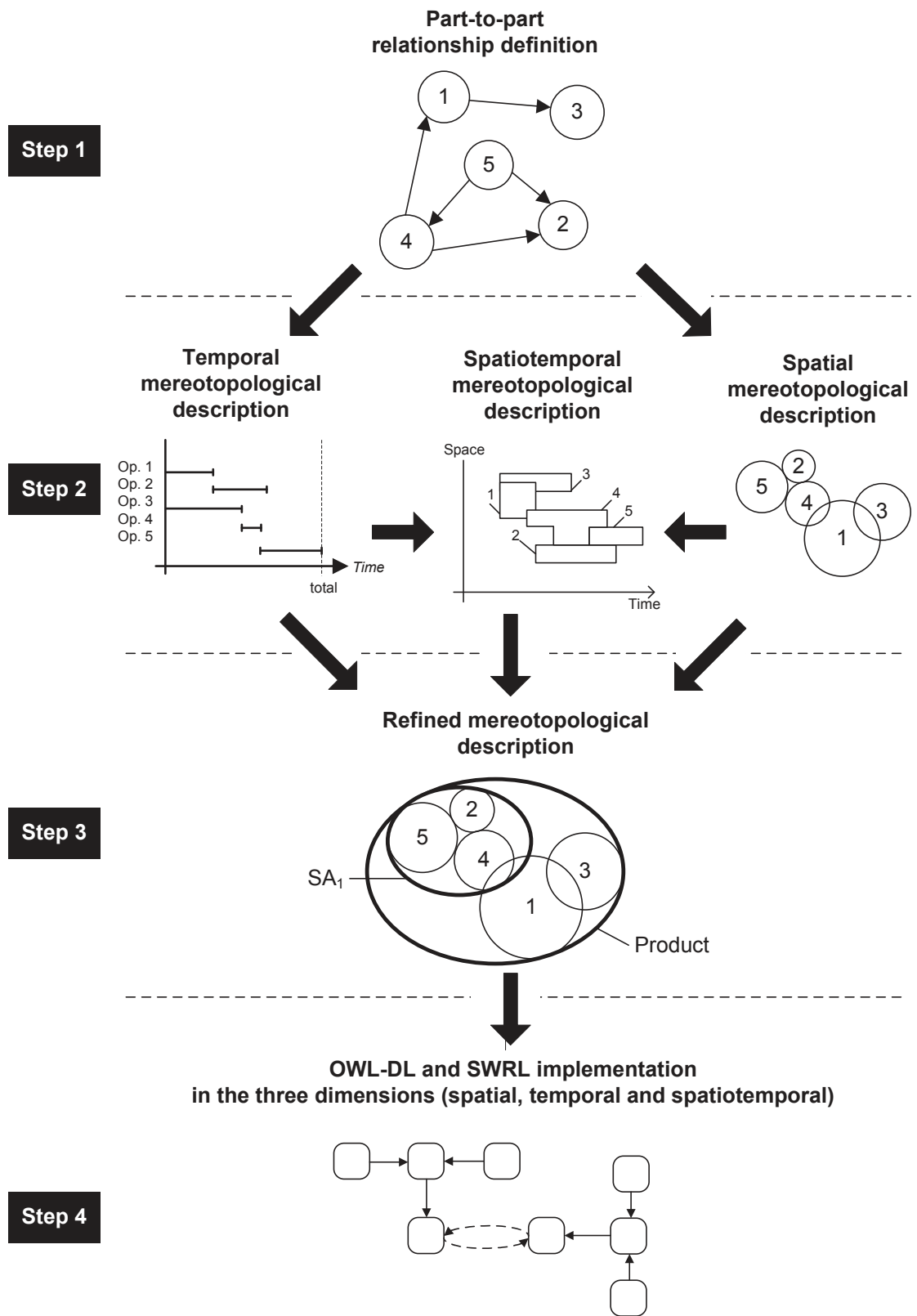


Figure 35: Mereotopological product relationships description approach (PRONOIA2) over space and time

The overall ontology is described in Figure 37. Each color represents a part of the ontology and is described in details in the next sections. The domain-ontology is divided into six groups (i.e. regions and also primitives each time in the three dimensions: spatial, temporal and spatiotemporal) so that the relationships between classes can be better visualized. The ontology is composed of terms derived from mereotopology (e.g. region, primitive and so on) and engineering design (e.g. part, interface skeleton and so on). However, all classes are directly related to the notions used in the JANUS mereotopological theory [Gruhier et al., 2014c].

### 4.3 Design rationale

The proposed ontological development is based on perdurantism or perdurance theory (also referred as four-dimensionalism [Sider, 2001]) in order to cover the description of product evolution in product design as well as in assembly planning phases [Gruhier et al., 2014c]. Here perdurantism can be considered as the antipode to endurantism, which presents entity as wholly present at every moment of its existence. In the proposed context, the perdurantist view considers that the product has distinct temporal parts throughout its existence, so called *timeslices* (i.e. entities representation only valid at time  $t$ ). As such, the series of all *timeslices* is associated to the full product description.

Built on this, the Methontology framework [Fernandez et al., 1997] has been used to provide a step-by-step guidance in the development of PRONOIA2 ontology. In the ontology development process – such as composed of specification, knowledge capture, conceptualization, formalization and implementation phases – the authors have considered JANUS theory as a strong input for each step of the Methontology method.

### 4.4 Meta-ontology description

This section introduces the meta-ontology representing objects and their relationships in the three dimensions: spatial, temporal and spatiotemporal. Classes, sub-classes and properties are described to build the knowledge base and illustrated in Figure 38. The model is composed of two top-level types of classes: the physical entity (i.e. region) and the relationship between these entities (i.e. primitive). Spatial and temporal regions and primitives are four different sub-classes of the meta-ontology.

As the spatiotemporal dimension depends on the spatial and temporal dimensions, it has been considered as a sub-class of those two dimensions. Classes and sub-classes are linked with properties, which are semantic relationships to facilitate the reuse of existing data, find the inconsistency and errors in data and aids designers to make right decisions by considering complex criteria [Chang, 2008].

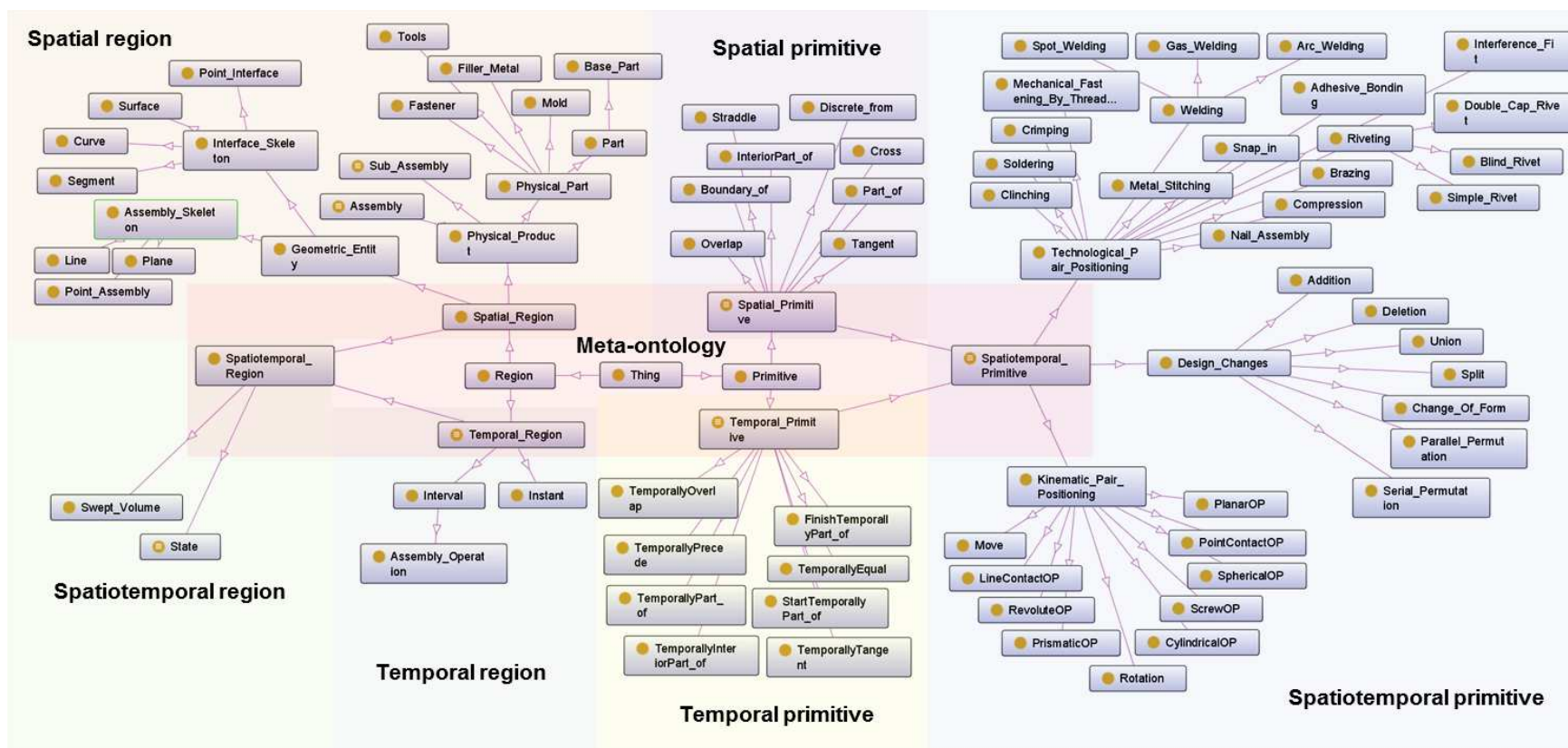


Figure 37: Overall view of PRONOA2 ontology



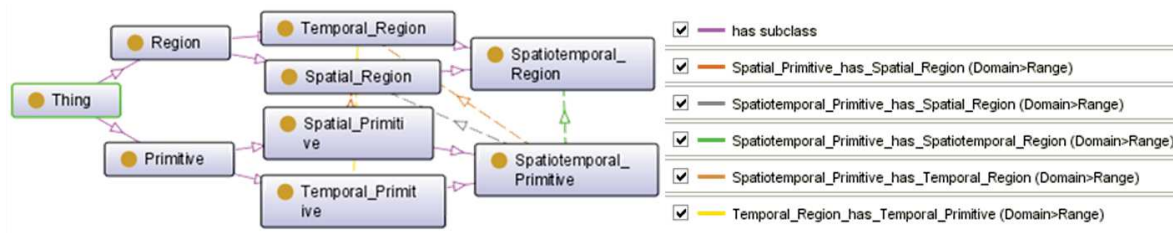


Figure 38: Classes, sub-classes and properties of the meta-ontology (PRONOA2) [Gruhier et al., 2014d]

## 4.5 Assembly-ontology description

Based on the meta-ontology, an assembly-ontology is defined in the AOD domain. Figure 39, Figure 40 and Figure 41 respectively show classes related to spatial, temporal and spatiotemporal regions. On the same way, Figure 42, Figure 43 and Figure 44 respectively present classes related to spatial, temporal and spatiotemporal primitives, in which sub-classes are associated to mereotopological primitives such as described in [Gruhier et al., 2014c]. Indeed, such primitives present interactions and contacts, that can occur for instance between two parts during the assembly process.

Here this ontology has specific vocabulary linked to the assembly process (such as swept volume), which differentiates it with other domain-ontologies (e.g. manufacturing-ontology). It also integrates the notion of topology with skeleton entities such as line, plane and point to name a few. Therefore spatiotemporal primitives (such as kinematic pairs positioning, technological pairs positioning and design changes), which are generated during the assembly process and design revisions, are formalized in this domain ontology. As such, all kinds of mechanical assembly can be described here.

The domain-ontology has been built in a way that, at a specific state of the assembly design, two kinds of situation can occur (cf. Figure 45). The first case is that at a specific temporal region, two spatial regions are linked with a spatial primitive. The second one is that at a specific temporal region, spatiotemporal region and spatial region are related each other with a spatiotemporal primitive. New introduced classes and sub-classes are represented in bold in Figure 46.

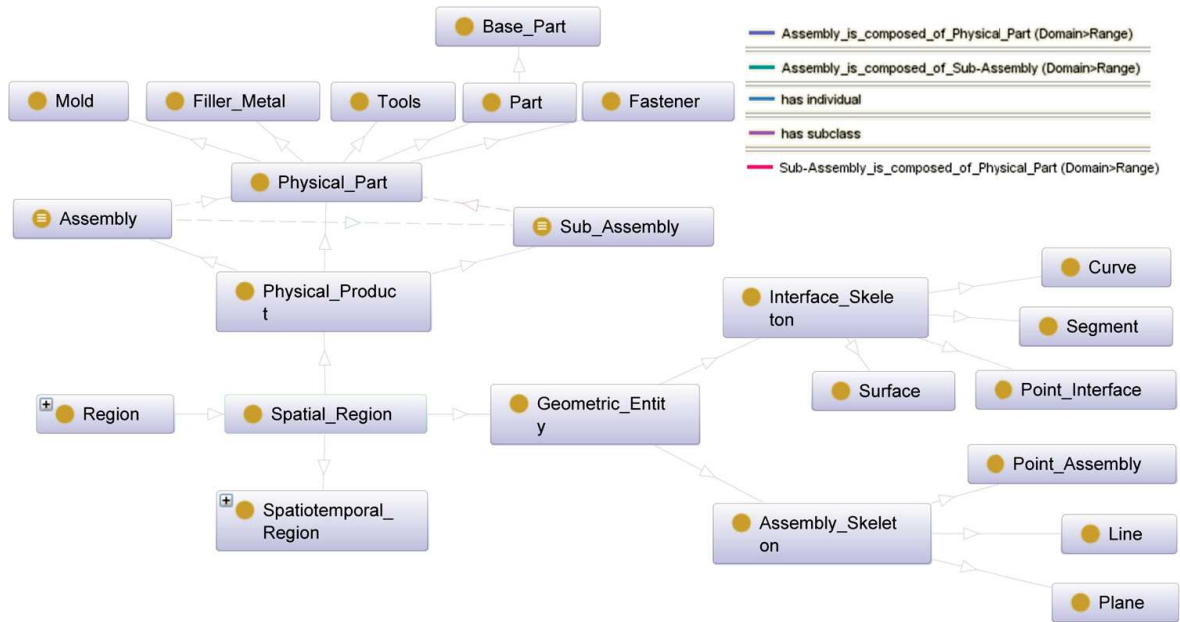


Figure 39: Classes and sub-classes related to spatial regions of the domain-ontology (PRONIA2) [Gruhler et al., 2015a]

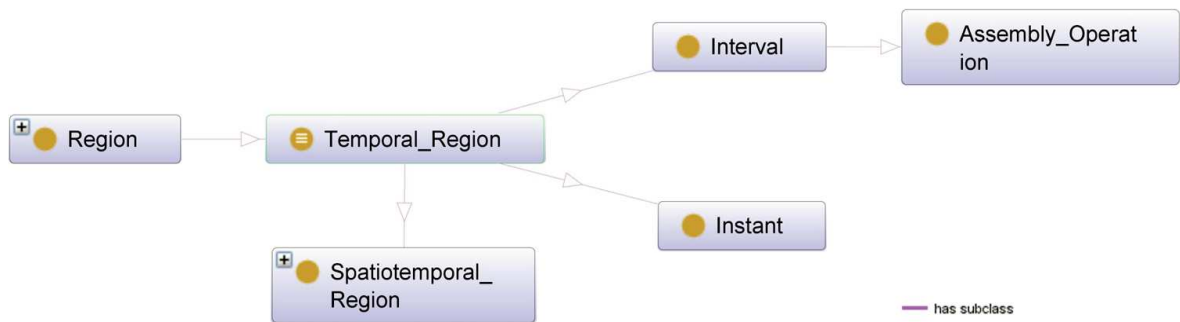


Figure 40: Classes and sub-classes related to temporal regions of the domain-ontology (PRONIA2) [Gruhler et al., 2015a]



Figure 41: Classes and sub-classes related to spatiotemporal regions of the domain-ontology (PRONOIA2)

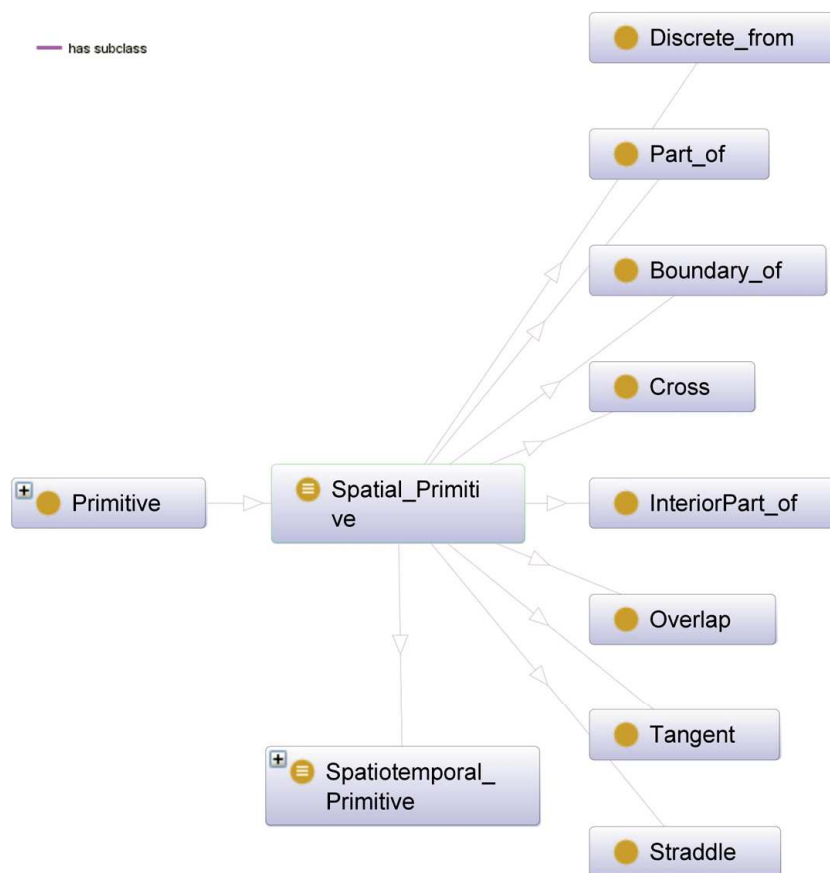


Figure 42: Classes and sub-classes related to spatial primitives of the domain-ontology (PRONOIA2) [Gruhler et al., 2015a]

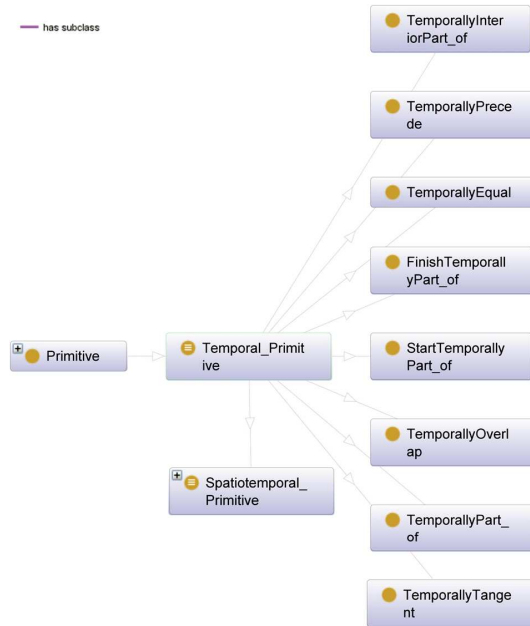


Figure 43: Classes and sub-classes related to temporal primitives of the domain-ontology (PRONIA2) [Gruhier et al., 2015a]

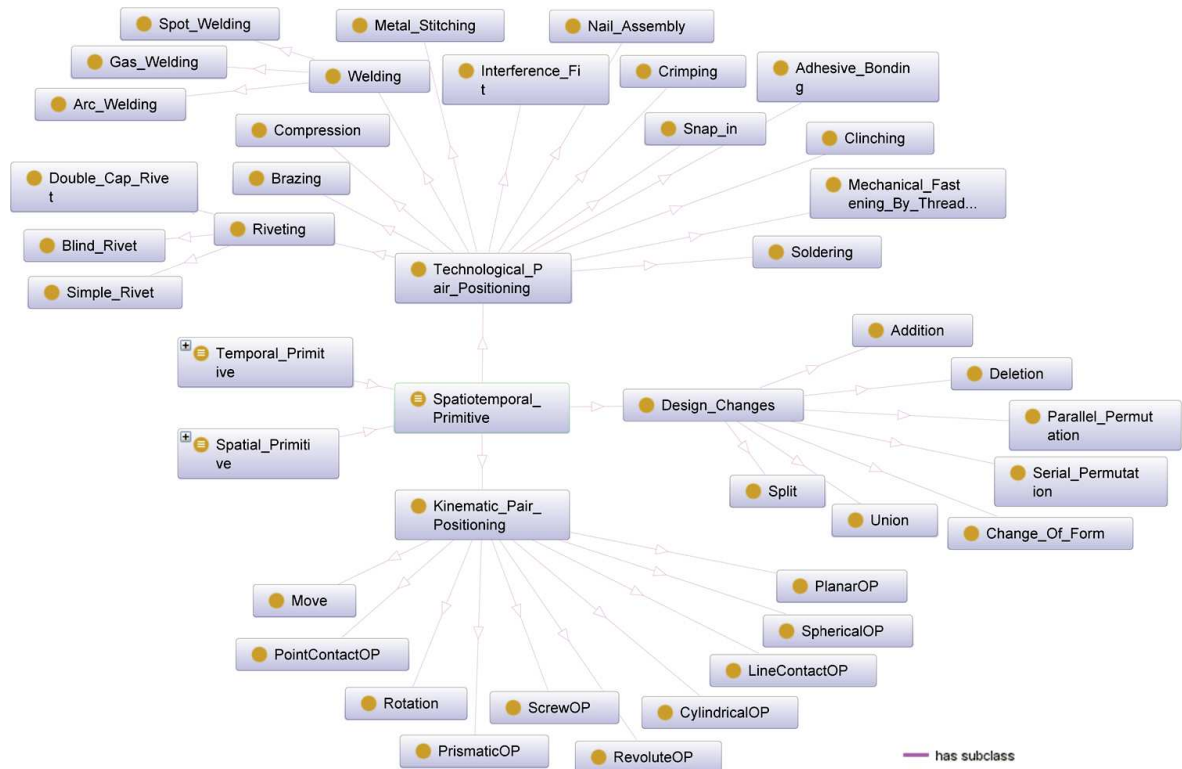


Figure 44: Classes and sub-classes related to spatiotemporal primitives of the domain-ontology (PRONIA2) [Gruhier et al., 2015a]

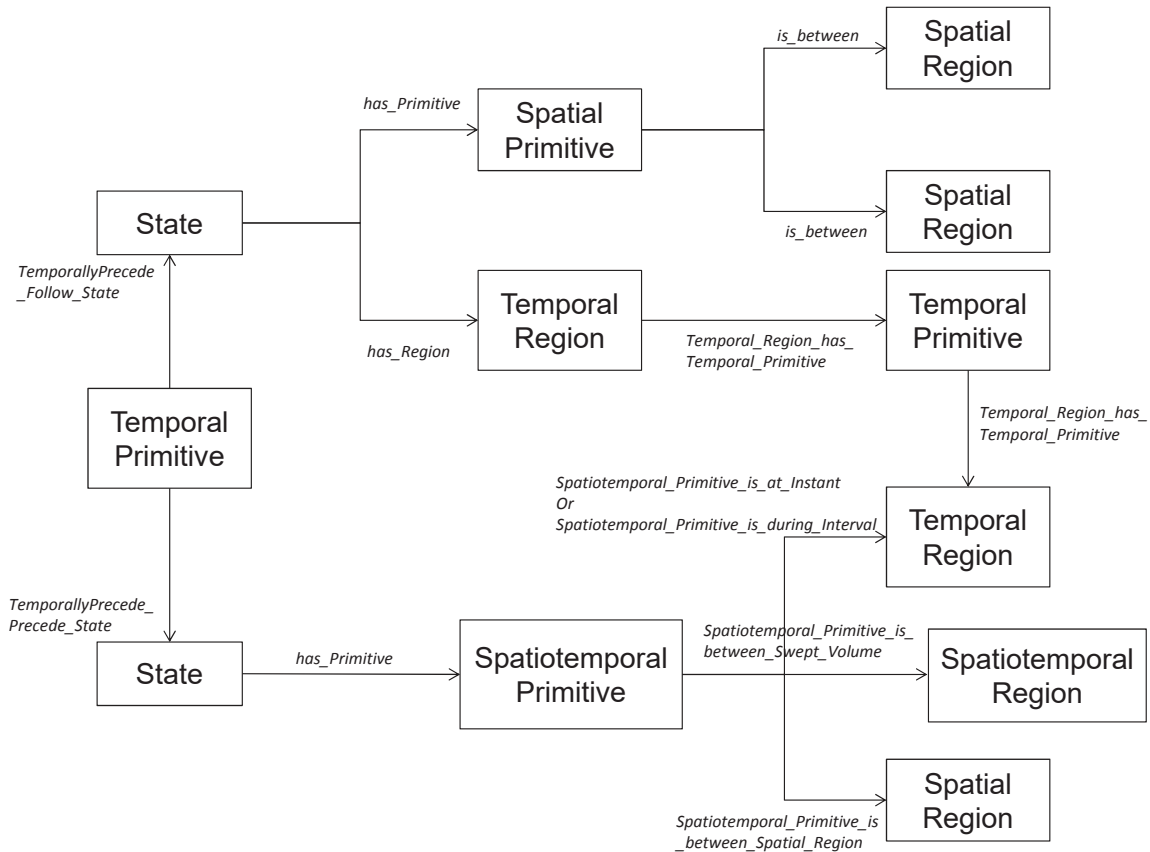


Figure 45: Building of the domain-ontology

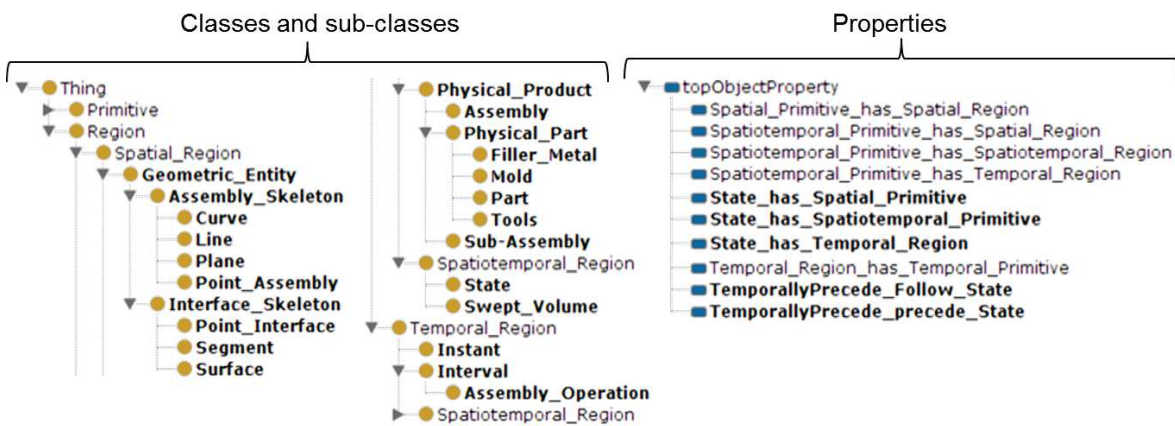


Figure 46: Classes, sub-classes and properties of the domain-ontology (PRONOIA2) [Gruhier et al., 2014d]

## 4.6 OWL-DL restriction

### 4.6.1 General information on OWL-DL

DL is a knowledge representation language used to represent knowledge of a specific domain in a structured manner. Intrinsic requirements (i.e. model structure, material, etc.)

are also expressed by DL and implied on classes [Fiorentini et al., 2010]. DL ontologies can identify conflicting knowledge and recognize the effects of changing information [Witherell et al., 2013]. As such, DL is used for information representation (e.g. `is-a` properties) and for inference mechanisms. Furthermore, consistency checking, subsumption, realization and retrieving are four kinds of reasoning on DL languages. Consistency checking ensures the coherence of the model. One way to check consistency is to verify that a class is not empty regarding properties and rules (i.e. `has individuals`).

OWL Lite, OWL-DL and OWL Full are three different languages. OWL Lite is the less expressive one and provides classes hierarchy and allows applying cardinality constraints. OWL-DL is a decidable language, that means it is possible to reason with adapted algorithms on these languages [Fortineau et al., 2013]. Finally, OWL Full is the most expressive language and is not decidable. At this stage, OWL is the most appropriate language to implement description logic needed for product modeling [Fiorentini et al., 2008]. OWL-DL expressivity is *SHOIN*. The DL concepts in OWL-DL are classes, the DL roles are object-properties and the DL individuals are instances. The top concept is the class “thing” [Fiorentini et al., 2008].

Classes and sub-classes are linked with properties, which are semantic relationships. The use of semantics facilitates the reuse of existing data, the finding of inconsistency and the assistance for designers to make right decisions by considering complex criteria [Chang, 2008]. Semantics provide structured relationships between entities of information and a way to add depth to information through systems and lifecycles [Witherell et al., 2013]. Besides, the more a class is restricted, the more it can be distinguished from others and be the only one. On the contrary, if a class has just few restrictions, it can be easily gathered with others which have common restrictions during the inference process.

Here, the following naming conventions are used as follows: the names of classes are written in capitalized/lower case Arial (i.e. `Physical_Product`), the names of attributes and relationships are capitalized/lower case Courier New (i.e. `isParentOf`) while names of instances are in italics Arial (i.e. *Physical\_Part\_1*) [Horrocks et al., 2007].

PRONOIA2 aims at assisting product architects and designers in the AOD process. The use of ontologies enables the development of knowledge-based systems, capable of reasoning by their own, via an inference engine embedded. Here, HermiT 1.3.7 reasoner, available in Protégé software, has been used to check the consistency of the ontology model [Horrocks et al., 2012]. Unlike Pellet reasoner which uses tableau calculus (built on different set of derivation rules), HermiT is based on hypertableau calculus. This approach allows the reasoner to avoid nondeterministic behavior with logic-based reasoning. HermiT is fully compliant with OWL2 language. Class and property can be classified with the subsumption algorithm that greatly reduces the number of consistency tests. Moreover SPARQL query answering and SWRL rules are also supported by HermiT [Glimm et al., 2014].

The ontology could lack consistency in the spatial dimension (e.g. an assembly is

Table 24: PRONOA2 DL rules in the meta-ontology

Class name	Rule type	DL Rule
Spatial_Primitive	Cardinality	Spatial_Primitive $\equiv$ is_between = 2 Spatial_Region
Temporal_Region	Some values from	Temporal_Region $\equiv$ has_Temporal_Primitive $\exists$ Temporal_Primitive

composed of only one part), temporal dimension (e.g. two temporal regions are linked with spatial primitives) or spatiotemporal dimension (e.g. *PrismaticOP* is composed of a line and a plane as geometric entity). The reasoner classifies and asserts classes and object property (cf. Figure 47).

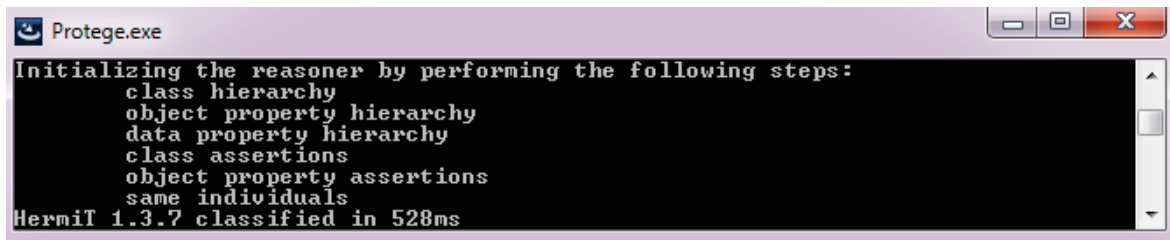


Figure 47: Result of consistency checking of PRONOA2 within Protégé using Hermit 1.3.7

#### 4.6.2 Restriction in the meta- and assembly-ontology

Table 24, Table 25, Table 26, Table 27 and Table 28 define DL rules which have been implemented in PRONOA2. Properties, which are used to describe restrictions, are written in the right ontology (i.e. if the property concerns classes in the meta-ontology, then property will be written in the related file). The objective is to write the restrictions as high as possible in the class hierarchy and approach the top concept to avoid repetition. That is the reason why Table 24 is composed of restrictions belonging to the meta-ontology. All other tables belong to the assembly-ontology.



Table 25: PRONOIA2 DL rules in the assembly-ontology

Class name	Rule type	DL Rule
Assembly	Min cardinality	$\text{Assembly} \equiv ((\text{is\_composed\_of\_Physical\_Part} \geq 1 \text{ Physical\_Part}) \sqcap (\text{is\_composed\_of\_Sub\_Assembly} \geq 1 \text{ Sub\_Assembly})) \sqcup (\text{is\_composed\_of\_Physical\_Part} \geq 2 \text{ Physical\_Part})$
Sub_Assembly	Min cardinality	$\text{Sub\_Assembly} \equiv \text{is\_composed\_of\_Physical\_Part} \geq 2 \text{ Physical\_Part}$
State	Min cardinality	$\text{State} \equiv ((\text{has\_Primitive} = 1 \text{ Spatiotemporal\_Primitive}) \sqcup (\text{has\_Region} = 1 \text{ Temporal\_Region})) \sqcap (\text{has\_Primitive} \geq 1 \text{ Spatial\_Primitive})$
Spatiotemporal_Primitive	Cardinality	$\text{Spatiotemporal\_Primitive} \equiv (\text{is\_at\_Instant} = 1 \text{ Instant}) \sqcup (\text{is\_during\_Interval} = 1 \text{ Interval})$
Spatiotemporal_Primitive	Cardinality	$\text{Spatiotemporal\_Primitive} \equiv ((\text{is\_between\_Spatial\_Region} = 1 \text{ Spatial\_Region}) \sqcap (\text{is\_between\_Swept\_Volume} = 1 \text{ Swept\_Volume})) \sqcup (\text{is\_between\_Spatial\_Region} = 2 \text{ Spatial\_Region})$
Addition	Cardinality	$\text{Addition} \sqsubseteq (\text{Other\_Evolution}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$
Change_Of_Form	Cardinality	$\text{Change\_Of\_Form} \sqsubseteq (\text{Other\_Evolution}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface})$
Deletion	Cardinality	$\text{Deletion} \sqsubseteq (\text{Other\_Evolution}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$
Parallel_Permutation	Cardinality	$\text{Parallel\_Permutation} \sqsubseteq (\text{Other\_Evolution}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface})$
Serial_Permutation	Cardinality	$\text{Serial\_Permutation} \sqsubseteq (\text{Other\_Evolution}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface})$
Split	Cardinality	$\text{Split} \sqsubseteq (\text{Other\_Evolution}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface})$
Union	Cardinality	$\text{Union} \sqsubseteq (\text{Other\_Evolution}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface})$



Table 26: PRONOA2 DL rules in the assembly-ontology

Class name	Rule type	DL Rule
CylindricalOP	Cardinality	$\text{CylindricalOP} \sqsubseteq (\text{Kinematic\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$
LineContactOP	Cardinality	$\text{LineContactOP} \sqsubseteq (\text{Kinematic\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Segment})$
Move	Cardinality	$\text{Move} \sqsubseteq (\text{Kinematic\_Pair}) \sqcap (((\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Plane})) \sqcup ((\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Point\_Assembly})) \sqcup ((\text{has\_Geometric\_Entity} = 1 \text{ Plane}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Point\_Assembly})) \sqcup (\text{has\_Geometric\_Entity} = 2 \text{ Line}) \sqcup (\text{has\_Geometric\_Entity} = 2 \text{ Plane}) \sqcup (\text{has\_Geometric\_Entity} = 2 \text{ Point\_Assembly}))$
PlanarOP	Cardinality	$\text{PlanarOP} \sqsubseteq (\text{Kinematic\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Plane}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$
PointContactOP	Cardinality	$\text{PointContactOP} \sqsubseteq (\text{Kinematic\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Point\_Assembly}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Point\_Interface})$
PrismaticOP	Cardinality	$\text{PrismaticOP} \sqsubseteq (\text{Kinematic\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$
RevoluteOP	Cardinality	$\text{RevoluteOP} \sqsubseteq \text{Kinematic\_Pair} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Plane}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$
Rotation	Cardinality	$\text{Rotation} \sqsubseteq (\text{Kinematic\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Point\_Assembly})$
ScrewOP	Cardinality	$\text{RevoluteOP} \sqsubseteq \text{Kinematic\_Pair} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Curve}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$
SphericalOP	Cardinality	$\text{SphericalOP} \sqsubseteq (\text{Kinematic\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Point\_Assembly}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$

Table 27: PRONOA2 DL rules in the assembly-ontology

Class name	Rule type	DL Rule
Adhesive_Bonding	Cardinality	$\text{Adhesive\_Bonding} \sqsubseteq (\text{Technological\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 3 \text{ Plane}) \sqcap (\text{has\_Geometric\_Entity} = 3 \text{ Surface})$
Brazing	Cardinality	$\text{Brazing} \sqsubseteq (\text{Technological\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Segment})$
Clinching	Cardinality	$\text{Clinching} \sqsubseteq \text{Technological\_Pair} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface}) \sqcap (\text{has\_Geometric\_Entity} = 3 \text{ Plane})$
Compression	Cardinality	$\text{Compression} \sqsubseteq \text{Technological\_Pair} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface}) \sqcap (\text{has\_Geometric\_Entity} = 4 \text{ Plane})$
Crimping	Cardinality	$\text{Crimping} \sqsubseteq (\text{Technological\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Plane})$
Interference_Fit	Cardinality	$\text{Interference\_Fit} \sqsubseteq (\text{Technological\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Plane})$
Mechanical_Fastening	Cardinality	$\text{Mechanical\_Fastening} \sqsubseteq \text{Technological\_Pair} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface}) \sqcap (\text{has\_Geometric\_Entity} = 3 \text{ Plane})$
Metal_Stitching	Cardinality	$\text{Metal\_Stitching} \sqsubseteq (\text{Technological\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 3 \text{ Plane}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface})$
Nail_Assembly	Cardinality	$\text{Nail\_Assembly} \sqsubseteq \text{Technological\_Pair} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface}) \sqcap (\text{has\_Geometric\_Entity} = 3 \text{ Plane})$
Snap_in	Cardinality	$\text{Snap\_in} \sqsubseteq \text{Technological\_Pair} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Plane})$
Soldering	Cardinality	$\text{Soldering} \sqsubseteq (\text{Technological\_Pair}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Segment})$

Table 28: PRONOA2 DL rules in the assembly-ontology

Class name	Rule type	DL Rule
Blind_Rivet	Cardinality	$\text{Blind\_Rivet} \sqsubseteq \text{Riveting} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Surface}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Plane})$
Double_Cap_Rivet	Cardinality	$\text{Double\_Rivet} \sqsubseteq \text{Riveting} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface}) \sqcap (\text{has\_Geometric\_Entity} = 4 \text{ Plane})$
Simple_Rivet	Cardinality	$\text{Simple\_Rivet} \sqsubseteq \text{Riveting} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface}) \sqcap (\text{has\_Geometric\_Entity} = 3 \text{ Plane})$
Arc_Welding	Cardinality	$\text{Arc\_Welding} \sqsubseteq (\text{Welding}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Plane})$
Gas_Welding	Cardinality	$\text{Gas\_Welding} \sqsubseteq (\text{Welding}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Segment})$
Spot_Welding	Cardinality	$\text{Spot\_Welding} \sqsubseteq \text{Welding} \sqcap (\text{has\_Geometric\_Entity} = 1 \text{ Line}) \sqcap (\text{has\_Geometric\_Entity} = 2 \text{ Surface}) \sqcap (\text{has\_Geometric\_Entity} = 3 \text{ Plane})$
Temporal_Primitive	Cardinality	$\text{Temporal\_Primitive} \equiv ((\text{is\_Between\_Instant} = 1 \text{ Instant}) \sqcap (\text{is\_Between\_Interval} = 1 \text{ Interval})) \sqcup (\text{is\_Between\_Instant} = 2 \text{ Instant}) \sqcup (\text{is\_Between\_Interval} = 2 \text{ Interval}) \sqcup (\text{is\_Between\_State} = 2 \text{ State})$

## 4.7 SWRL rules

SWRL is the combination of OWL-DL and **RuleML** (**R**ule **M**arkup **L**anguage). RuleML is based on implications. SWRL rules are introduced, since they provide a greater expressivity than DL ones [Fiorentini et al., 2008], which cannot consider abstract requirements regarding the product (i.e. classification, application, etc.) [Fiorentini et al., 2010].

A typical SWRL rule is used to express that the conjunction of various causes implies the establishment of a new property between instances (If ..., then ...). Therefore SWRL increases the inferring of new knowledge [Fortineau et al., 2013]. Table 29 presents the SWRL rules describing the final state of the kinematic pairs between two parts with skeletons.

One specific SWRL rule, *Revolutes*, is defined in details. This rule says that for two parts (i.e.  $x$  and  $y$ ) being in revolute operation, they are crossing and are both overlapping the skeleton line (i.e.  $ka$ ). Besides, they are tangent to the skeleton plane (i.e.  $kb$ ) and tangent to the interface skeleton (i.e.  $f$ ). SWRL allows explicitly representing constraints and bringing some logic in the management of product-process information through PLM systems.

Table 29: SWRL rules in PRONOA2 ontology

SWRL Rules
$\text{Overlaps}(?x, ?k) \wedge \text{Overlaps}(?y, ?k) \wedge \text{Crosses}(?x, ?y) \wedge \text{has\_line}(?x, ?k) \wedge \text{has\_line}(?y, ?k) \wedge \text{has\_surface}(?x, ?f) \wedge \text{has\_surface}(?y, ?f) \wedge \text{is\_Tangent\_of}(?x, ?f) \wedge \text{is\_Tangent\_of}(?y, ?f) \wedge \text{DifferentFrom}(?x, ?y) \rightarrow \mathbf{Cylindrical}(?x, ?y) \vee \mathbf{Prismatic}(?x, ?y)$
$\text{Straddles}(?x, ?y) \wedge \text{InteriorPart}(?x, ?y) \wedge \text{Overlaps}(?x, ?k) \wedge \text{Overlaps}(?y, ?k) \wedge \text{has\_point}(?x, ?k) \wedge \text{has\_point}(?y, ?k) \wedge \text{has\_surface}(?x, ?f) \wedge \text{has\_surface}(?y, ?f) \wedge \text{is\_Tangent\_of}(?x, ?f) \wedge \text{is\_Tangent\_of}(?y, ?f) \wedge \text{DifferentFrom}(?x, ?y) \rightarrow \mathbf{Spherical}(?x, ?y)$
$\text{Overlaps}(?x, ?ka) \wedge \text{Overlaps}(?y, ?kb) \wedge \text{Crosses}(?x, ?y) \wedge \text{has\_line}(?x, ?ka) \wedge \text{has\_line}(?y, ?ka) \wedge \text{has\_plane}(?x, ?kb) \wedge \text{has\_plane}(?y, ?kb) \wedge \text{has\_surface}(?x, ?f) \wedge \text{has\_surface}(?y, ?f) \wedge \text{is\_Tangent\_of}(?x, ?f) \wedge \text{is\_Tangent\_of}(?y, ?f) \wedge \text{is\_Tangent\_of}(?x, ?kb) \wedge \text{is\_Tangent\_of}(?y, ?kb) \wedge \text{DifferentFrom}(?x, ?y) \rightarrow \mathbf{Revolutes}(?x, ?y)$
$\text{is\_Tangent\_of}(?x, ?y) \wedge \text{has\_plane}(?x, ?k) \wedge \text{has\_plane}(?y, ?k) \wedge \text{has\_surface}(?x, ?f) \wedge \text{has\_surface}(?y, ?f) \wedge \text{is\_Tangent\_of}(?x, ?k) \wedge \text{is\_Tangent\_of}(?y, ?k) \wedge \text{is\_Tangent\_of}(?x, ?f) \wedge \text{is\_Tangent\_of}(?y, ?f) \wedge \text{DifferentFrom}(?x, ?y) \rightarrow \mathbf{Planar}(?x, ?y)$
$\text{is\_Tangent\_of}(?x, ?y) \wedge \text{has\_plane}(?x, ?k) \wedge \text{has\_plane}(?y, ?k) \wedge \text{has\_point}(?x, ?f) \wedge \text{has\_point}(?y, ?f) \wedge \text{is\_Tangent\_of}(?x, ?k) \wedge \text{is\_Tangent\_of}(?y, ?k) \wedge \text{is\_Tangent\_of}(?x, ?f) \wedge \text{is\_Tangent\_of}(?y, ?f) \wedge \text{DifferentFrom}(?x, ?y) \rightarrow \mathbf{PointContact}(?x, ?y)$
$\text{is\_Tangent\_of}(?x, ?y) \wedge \text{has\_plane}(?x, ?k) \wedge \text{has\_plane}(?y, ?k) \wedge \text{has\_segment}(?x, ?f) \wedge \text{has\_segment}(?y, ?f) \wedge \text{is\_Tangent\_of}(?x, ?k) \wedge \text{is\_Tangent\_of}(?y, ?k) \wedge \text{is\_Tangent\_of}(?x, ?f) \wedge \text{is\_Tangent\_of}(?y, ?f) \wedge \text{DifferentFrom}(?x, ?y) \rightarrow \mathbf{LineContact}(?x, ?y)$
$\text{Crosses}(?x, ?y) \wedge \text{has\_line}(?x, ?k) \wedge \text{has\_line}(?y, ?k) \wedge \text{has\_curve}(?x, ?fb) \wedge \text{has\_curve}(?y, ?fb) \wedge \text{has\_curve}(?x, ?fc) \wedge \text{has\_curve}(?y, ?fc) \wedge \text{has\_surface}(?x, ?fa) \wedge \text{has\_surface}(?y, ?fa) \wedge \text{Overlaps}(?x, ?k) \wedge \text{Overlaps}(?y, ?k) \wedge \text{is\_Tangent\_of}(?x, ?fa) \wedge \text{is\_Tangent\_of}(?y, ?fb) \wedge \text{is\_Tangent\_of}(?x, ?fc) \wedge \text{Overlaps}(?fb, ?fc) \wedge \text{DifferentFrom}(?x, ?y) \wedge \text{DifferentFrom}(?fb, ?fc) \rightarrow \mathbf{Screw}(?x, ?y)$

## 4.8 Summary

Based on a strong state of the art survey on spatial, temporal and spatiotemporal ontology models, this chapter has introduced a research effort towards a spatiotemporal ontology in AOD. The proposed PRONOIA2 ontology is based on a mereotopological theory (JANUS), which qualitatively describes product-process definition and evolution. As a consequence, the ontology covers three dimensions, such as spatial, temporal and spatiotemporal, so as to check information consistency issues in PLM. Here, the spatiotemporal dimension has been introduced in order to be able to describe product-process knowledge and information in a consistent and understandable manner. This dimension aids designers to understand changes during AOD.

The PRONOIA2 ontology model is composed of a meta-ontology, a domain-ontology (in that case the AOD domain) and an application-ontology. Hence, the JANUS theory is now formalized and machine-interpretable by PLM systems, since an OWL implementation has been carried out. The actual stake is to get a long term dynamic vision of the space in order to facilitate the understanding of assembly and design changes. As such, a reasoning layer has been implemented so as to reason on spatiotemporal associations. Rules have been added using SWRL and DL languages so as to reason on spatiotemporal associations within PLM systems (i.e. PDM and MPM systems).

Such efforts will enable the introduction of novel procedures for consistency checking of product-process information and knowledge in PLM and create a robust epistemological and methodological platform. We conclude this chapter by the necessity to link PRONOIA2 ontology to information systems such as PDM and MPM, and CAD tools through a web service. As a consequence, product evolution will be managed and product architects and designers will be aware of design intents in the early design phase. Therefore the next chapter will introduce the proposed framework and address the information flow between information systems.



# Chapter 5

## A PLM-based approach to manage assembly and design evolutions – MERCURY

### Contents

---

<b>5.1</b>	<b>Introduction . . . . .</b>	<b>100</b>
<b>5.2</b>	<b>Approach objectives . . . . .</b>	<b>100</b>
<b>5.3</b>	<b>Two different applications: Assembly and design evolutions .</b>	<b>103</b>
5.3.1	Product evolution control at the early design stages . . . . .	103
5.3.2	Design evolution control during the design process . . . . .	104
<b>5.4</b>	<b>Management of product-process information consistency . .</b>	<b>104</b>
<b>5.5</b>	<b>Overall framework of MERCURY . . . . .</b>	<b>110</b>
<b>5.6</b>	<b>Overview of information flow through PLM . . . . .</b>	<b>113</b>
<b>5.7</b>	<b>Summary . . . . .</b>	<b>122</b>

---



## 5.1 Introduction

As seen previously, JANUS theory has been developed in the first step (cf. Figure 63) of the research work. It aims at describing regions and primitives in the three dimensions (i.e. spatial, temporal and spatiotemporal). As such, the product evolution is fully described in the context of AOD from a perdurantist point of view. Then, PRONOIA2 approach has been developed to formalize JANUS theory by using semantics. Moreover, some logical restrictions and rules have been added in order to reason and check information consistency. The next step (Step 3 in Figure 63) is to manage product relationships information flow through PLM.

In this chapter, an information and knowledge management layer is added to current developed PRONOIA2 ontology and JANUS theory (cf. Figure 48). PRONOIA2 ontology is linked to information systems (such as PDM and MPM systems), applications (such as CAD tools) through a hub system (i.e. Pegasus [Demoly et al., 2013a]). As a consequence, the designers' activities will be supported in the early design phase and he will be more aware of product architects' design intents. Product architects' activities will be likewise supported during the design process, as designers' intents will be known. As such, all conditions will be gathered to design a consistent product over space and time.

## 5.2 Approach objectives

The main objective of this chapter is to develop an approach – called MERCURY – to manage changes that undergoes a product throughout its design and assembly processes. As stated in previous sections, a product can change through space, time and form [Le Moigne, 1994]. As such, the approach has to support spatiotemporal information and bring a product-process articulation.

Table 30 presents how changes are currently managed and how the MERCURY approach considers changes. Changes are split according to their dimensions. Here the second row introduces change management in current information systems. Only documents are managed. A lack of temporal and spatiotemporal change management is highlighted. The third row is focused on the proposed approach, which considers changes each time a mereotopological primitive is modified. For instance, spatial changes occur if a spatial mereotopological primitive from JANUS theory is modified. Likewise, form changes are detected if design changes primitives (such as *Growing*, *Deformation* and so on) are used. Here information is managed. As such, the novel approach is based on relationships.

Despite the fact that change management is already taken into consideration by some information systems, in current industrial project context, PDM, MPM and CAD are working separately. Hence, no change can be transmitted to another information systems and this can lead to an inconsistent designed product. So here the issue is how to control information flow and change management between PDM, MPM and CAD (cf. Figure 49).

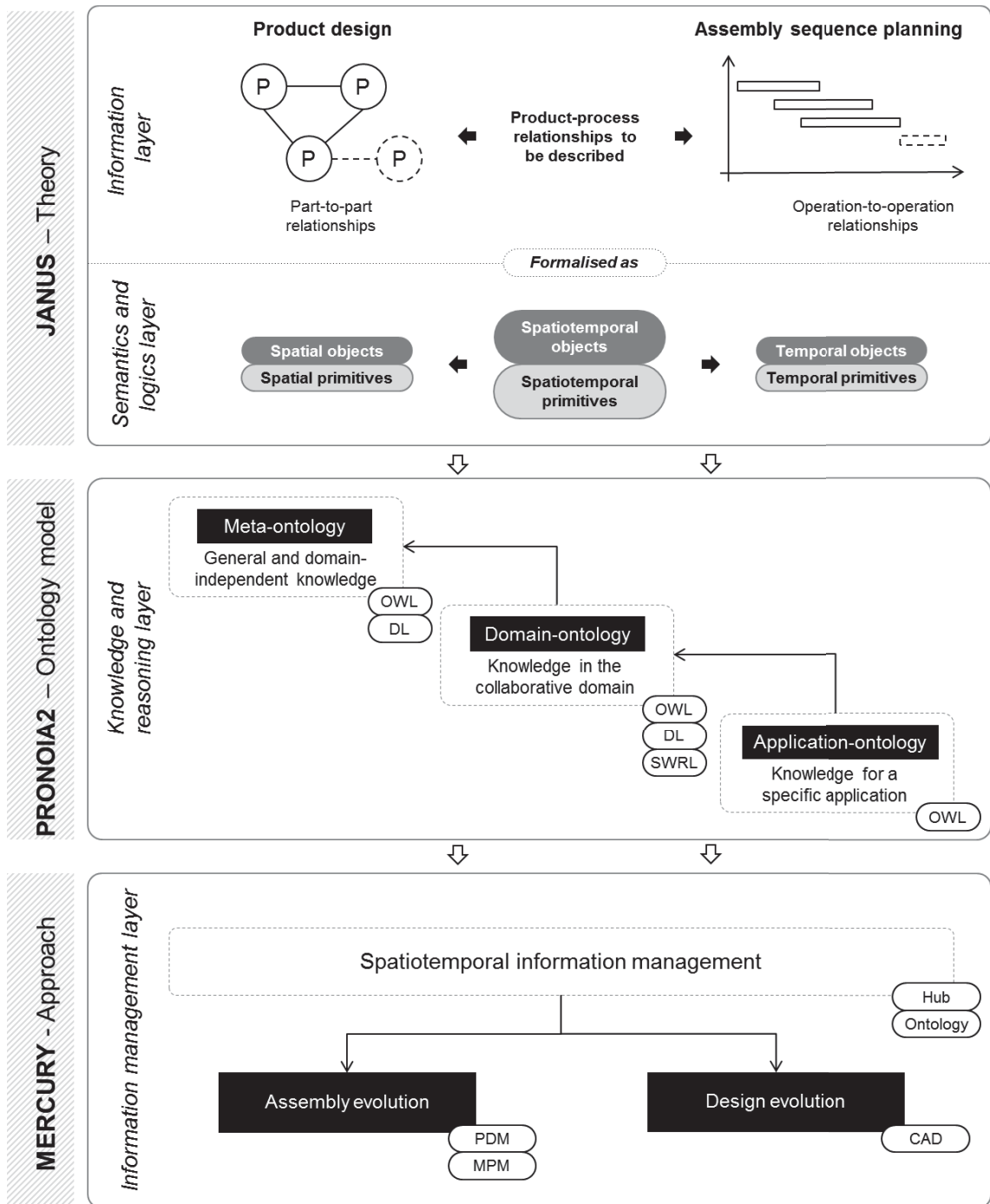


Figure 48: Management approach based on PRONOA2 ontology and JANUS theory

Table 30: Changes management and consideration

	Current change management	Change consideration with MERCURY
<b>Spatial change</b>	Files iteration in PDM with different versions	Spatial primitive
<b>Temporal change</b>		Temporal primitive
<b>Form change</b>	Files iterations	Design changes primitive
<b>Spatiotemporal change</b>		Kinematic and technological pairs positioning primitive

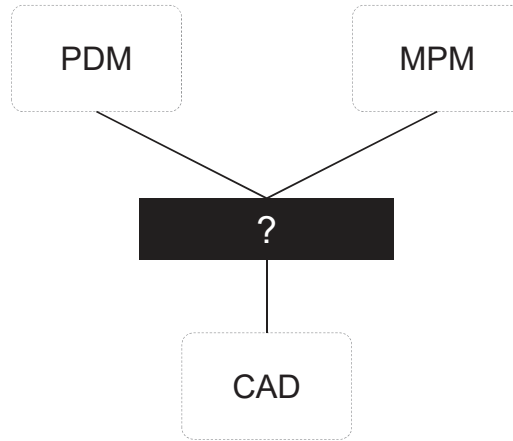


Figure 49: How to orchestrate information flow between PDM, MPM and CAD?

Here, the proposed MERCURY approach attempts to orchestrate information coming from these different systems so as to be able to deal with product, which dynamically evolves throughout its lifecycle.

Besides, a top-down approach is followed, as the whole product is considered before being studied in details with the mechanical parts. The approach aims at promoting a proactive and concurrent engineering. As such, the MERCURY approach objectives are to:

1. Involve different stakeholders (i.e. product architect, assembly planner and designer) working on the design and assembly phases in the early design stages;
2. Guarantee the consistency of PDM, MPM and CAD information;
3. Provide a context, defined by relational information, describing product evolution and supporting the product development process;
4. Manage product relationships between PDM, MPM systems and CAD tools;
5. Assure relationships between parts during the design process;
6. Improve designer's understanding on product architect's and assembly planner's intents, as well as improve product architect's understanding on designer's intents.

### 5.3 Two different applications: Assembly and design evolutions

The proposed approach can be applied on two different applications: to control product evolution at the early design stages and to control changes during the design process (cf. Figure 50). In the first application, product evolution mainly concerns the product move during the assembly process (e.g. kinematic and technological pairs positioning). On the contrary, changes deal with design evolution in the second application. Figure 51 presents a short example to understand both applications. The example is composed of three parts: Part 1 (i.e. shaft), Part 2 (i.e. ring) and Part 3 (i.e. ring).



Figure 50: Lifecycle positions of both applications

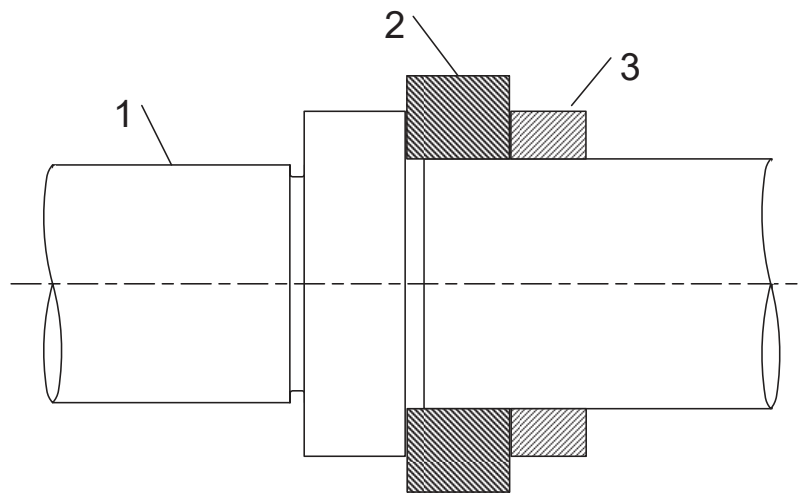


Figure 51: Mechanical assembly example for both applications

#### 5.3.1 Product evolution control at the early design stages

The first application describes the product evolution in the AOD context. The product architect, with the aid of the assembly planner, chooses the following assembly operations (cf. Figure 52):

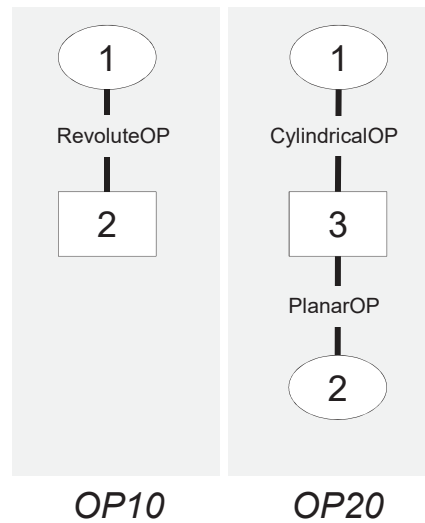


Figure 52: Representation of product evolution in the AOD context

- OP10: 1 *RevoluteOP* 2;
- OP20: 3 *CylindricalOP* 1  $\wedge$  3 *PlanarOP* 2.

In this application, concurrent and proactive engineering are ensured and designer is aware of product architect's design intents. Framework and information flow will be explained in details in next sections.

### 5.3.2 Design evolution control during the design process

The second application describes changes management during the design process. Here, the designer is developing the product and a traceability of his activities is saved. Changes occur because of designer experience. As such, product architect will be able to validate design and understand designer's intents. For example, the design changes operations are (cf. Figure 53):

- TR10: Modification of *Part 2* – 2 *Growing* 1;
- TR20: Modification of *Part 2* and *Part 3* – 2 *Union* 3.

Design changes, as considered here, cannot be controlled at the early design stages. Indeed, designers cannot be aware that a part should be grown before designing it.

## 5.4 Management of product-process information consistency

The research domain (cf. Figure 54) is located between PLM systems (i.e. PDM and MPM) and CAD applications. PDM and MPM have been distinctly developed with sepa-

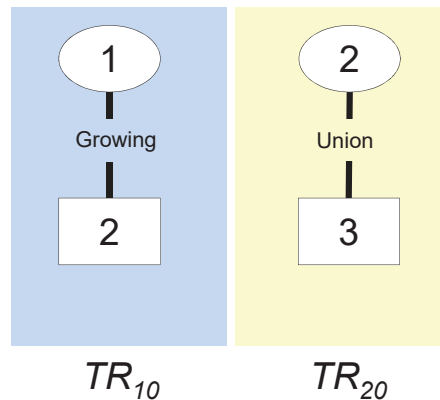


Figure 53: Representation of design changes

rately stored information, resulting in systems that are not well integrated [Peachavanish et al., 2006]. Indeed, current PDM systems only deal with spatial information (e.g. product component) and MPM system with temporal information (e.g. assembly operation). As such, spatial and temporal information are managed one at a time by two different stakeholders (i.e. respectively product architect and assembly planner), as product design and assembly process are considered as two distinct lifecycle phases of the product development. Therefore, a poor interaction between heterogenous information systems is highlighted.

Hence, a product-process management approach is needed in order to support concurrent engineering. The approach aims at managing the spatial and temporal information (corresponding to the product-process information). Indeed, a conductor is required to manage information coming from PDM and MPM. As such, a continuity between the assembly and design phases will occur and the designer will be able to work in a proactive context by integrating assembly sequence information in the preliminary design process.

In the context of AOD, two dimensions are considered: the spatial (e.g. mechanical part) and temporal (e.g. assembly operation) dimensions. Each dimension is controlled by one stakeholder. Product architect deals with spatial information and assembly planner with temporal information. Here the objective is to link spatial and temporal information through spatiotemporal relationships (cf. Figure 55). The management of these spatiotemporal relationships ensures a better interaction and coordination of product-process information [Brown, 2006]. These relationships will enable the information flow consistency in order to get a product, which can be assembled.

In another way, linking spatial and temporal information is like reconciliated PDM and MPM systems (cf. Figure 56). PDM usually deals with eBOM to manage product information. On the same way, MPM deals with mBOM and **BOP** (**B**ill **O**f **P**rocess) to manage process information. As such, the objective is to add spatiotemporal relationships at the early design stages, so as to improve designer's understanding. Indeed, the

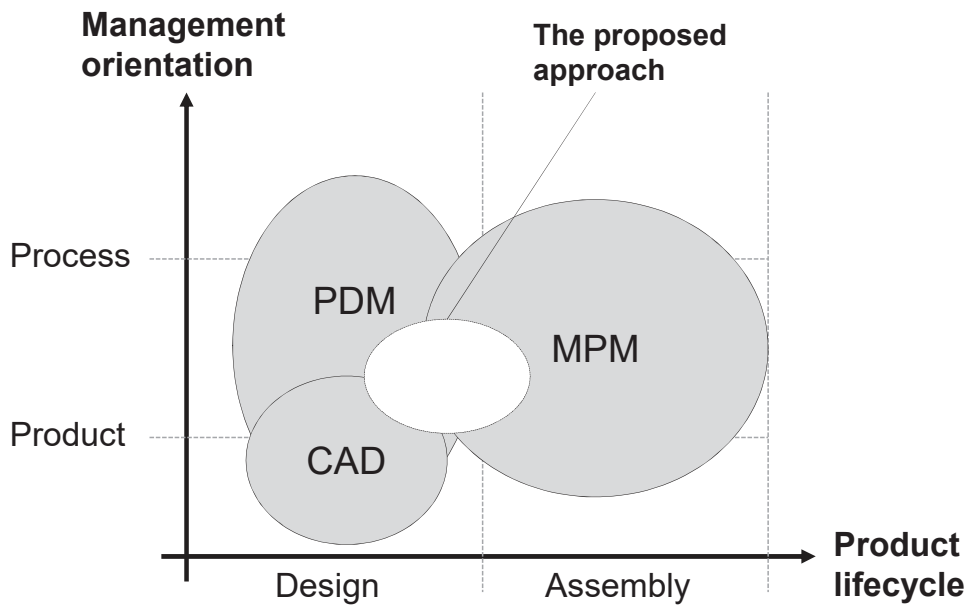


Figure 54: Research map of PLM systems and CAD tools

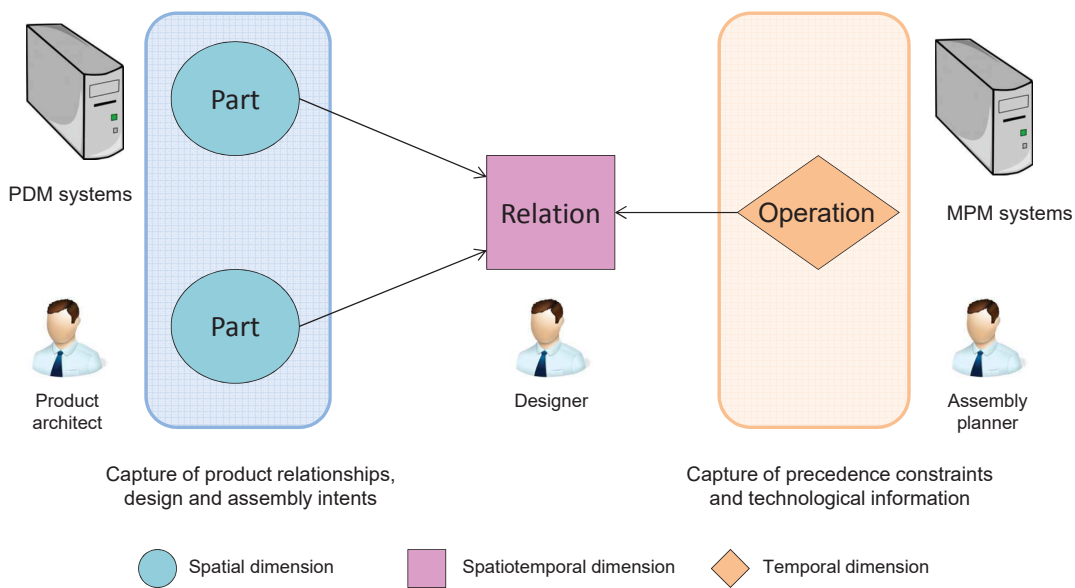


Figure 55: Link between spatial and temporal information through spatiotemporal relationships

spatiotemporal relationships bring new information to the designer such as the assembly constraints. Therefore, this information enables the designer to early consider these constraints. He will also be aware of design intents, design changes over time and information

which is usually hidden in PDM iteration files. As such, harmony will lead the assembly construction (i.e. study to know how parts are linked over space and time). This new approach, to link PDM and MPM systems, brings relief (i.e. with the new spatiotemporal dimension, which enables the dynamic product description and which is closer to the reality) and depth (i.e. as the product evolution is more detailed than in current CAD file) to current design approach.

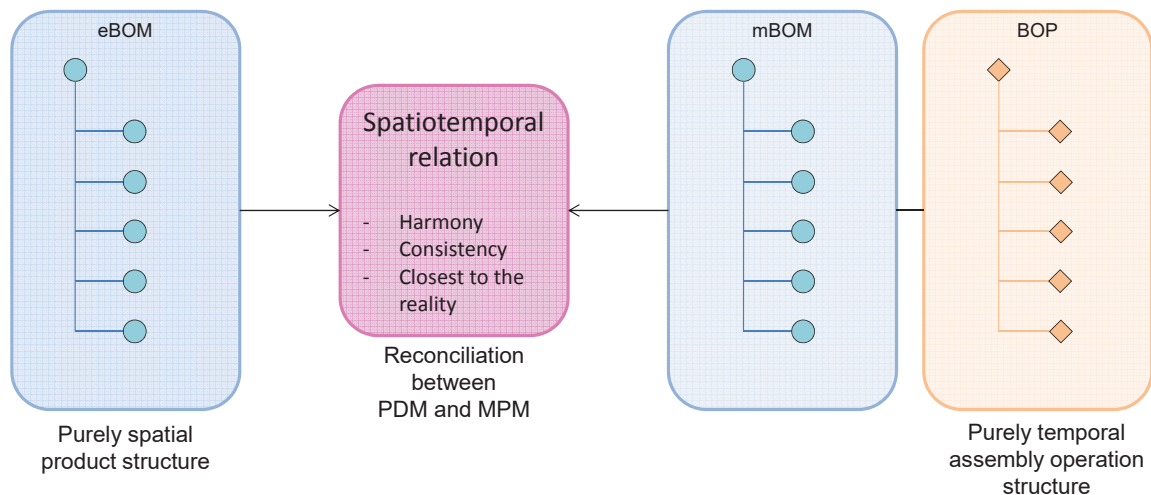


Figure 56: Reconciliation between PDM and MPM systems

Based on previous statements, product and process require to be linked in order to ensure concurrent product design (cf. Figure 56). The link is supported by product-process relationships. A relationship is considered as the means to establish, represent or maintain a consistent link between two technical entities [Demoly et al., 2012d]. As such, parts will be related to each other through the design and assembly phases. This kind of relationships has already been described in JANUS theory, especially with spatiotemporal mereotopological primitives. Moreover, product and process are currently managed in separate information systems (i.e. PDM and MPM) and by different stakeholders (i.e. product architect and assembly planner). As such, the MERCURY approach proposes to associate PDM and MPM systems so that stakeholders could work collaboratively in an AOD environment. The mathematical foundation has also been formalized in PRONOIA2 ontology. As such, the theory and the ontology enable adding a spatiotemporal layer to PLM systems, by capturing five kinds of relationships as described below (the first two relationships are based on Demoly's work [Demoly et al., 2012d]):

- Contact relation: Physical contact between two mechanical parts;
- Precedence relation: Assembly order between two mechanical parts, which are/are not in contact;



- Design change relation: Additional information on contact relation which enables the spatiotemporal description of the internal product evolution (e.g. *Growing*);
- Kinematic pair relation: Additional information on contact relation which enables the spatiotemporal description of the positioning of a part on another one according to the defined degree of freedom (e.g. *RevoluteOP*);
- Technological pair relation: Additional information on contact relation which enables the spatiotemporal description of the positioning of a part on another to assume the part function (e.g. *Arc Welding*).

In addition, three kinds of association gathering entities are proposed:

- Spatial link: describes a link between a part and its skeletons in product domain;
- Temporal link: describes a link between two assembly operations in assembly process domain;
- Spatiotemporal link: describes a link between two entities (i.e. a part and a swept volume) in the AOD domain.

The MUVOA model has been developed by Demoly et al. [Demoly et al., 2012d] and is supported by Pegasus. The objective of the model is to split the product according to the different viewpoints related to the different project stakeholders (i.e. product architect, assembly planner, designer and so on). This model highlights the relationships, as they enable knowing the design and assembly intents.

Based on the MUVOA model, Figure 57 presents the relationships between the views (e.g. structural, functional, contextual and so on) in the product and assembly process domain, and also between both domains. The black arrows are based on Demoly's work [Demoly et al., 2012d], the red ones are introduced to highlight our contribution and the green ones present the improved links compared to previous works. For instance, the approach have added a link between the geometric view of the product domain and the behavioral view in the assembly process domain. Above all, the approach brings a novel domain called product-process. Information in this domain is managed in a hub. In this specific domain, **BOR** (**B**ill **O**f **R**elations), swept volume and spatiotemporal relationships are controlled. As such, the approach reinforces the link between the product and process domains.

Table 31 summarizes the different views with their domains and the related stakeholders, which are in charge of this task. An example of the view is also presented in the last column. For instance, the contextual view with the assembly planner and the product architect in the product domain provides information on kinematic pairs, precedence and contact relations.

Therefore this MERCURY approach ensures a concurrent product design and assembly sequence planning with the management of product relationships. Moreover, the

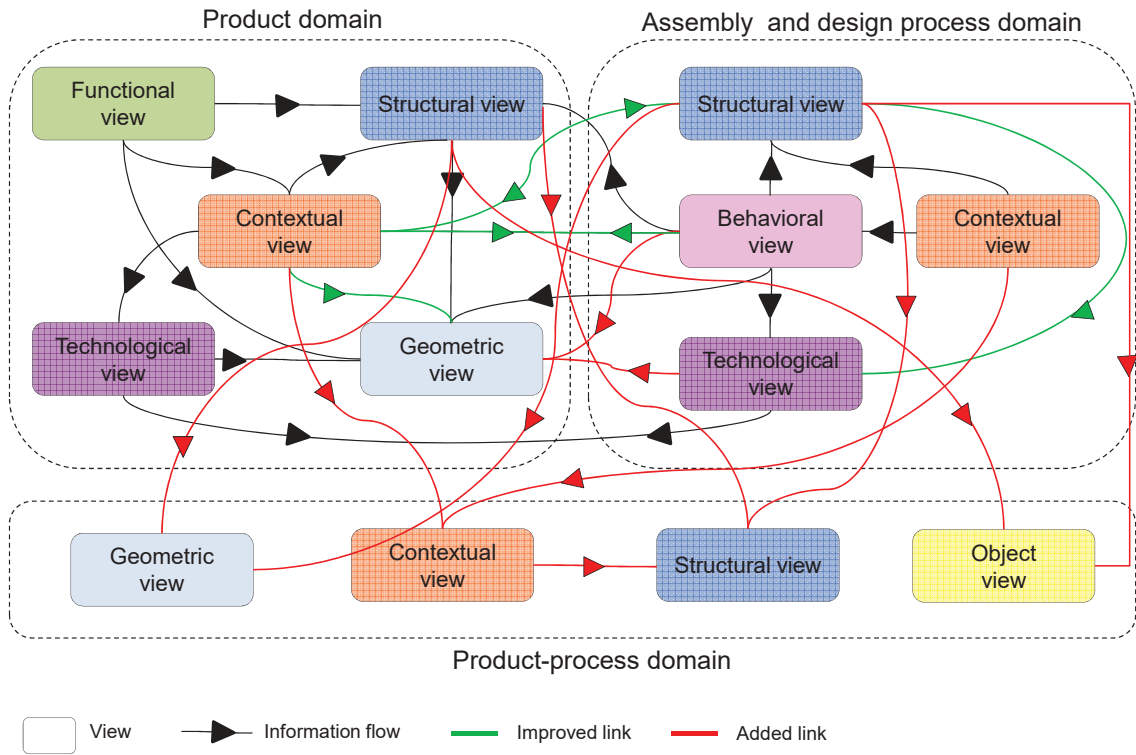


Figure 57: Comparison between the information flow in the MUVOA model [Demoly et al., 2012d] and the proposed model

Table 31: Link between stakeholders and views in product and assembly process

Domain	View	Stakeholder	Example of view
Product	Functional	Product architect	Technical functions
	Technological	Product architect & Designer	Technological pairs
	Structural	Product architect	Components configurations
	Geometric	Designer	Assembly constraints
	Contextual	Assembly planner & Product architect	Kinematic pairs, precedence and contact relations
Assembly & Design process	Structural	Assembly planner & Designer	Assembly and design operations
	Contextual	Assembly planner & Designer	Manufacturing and design context
	Behavioural	Assembly planner	Assembly sequence
	Technological	Assembly planner	Technological pairs
Product-process	Contextual	Product architect	Kinematic & technological pairs positioning
	Structural	Product architect & Designer	BOR
	Geometric	Designer	Design changes
	Object	Product architect	Swept volume

relationships management provides a better understanding of how parts are assembled or changed over time. As such, the information propagation through design and manufacturing structures is managed and will be detailed in the next section.

## 5.5 Overall framework of MERCURY

The simplified framework is presented in Figure 58. The colored box represents the spatial information managed by PDM, the temporal information in MPM, the information about the product form in CAD and especially the novel entities managed by the hub application by a mediator and an agent.

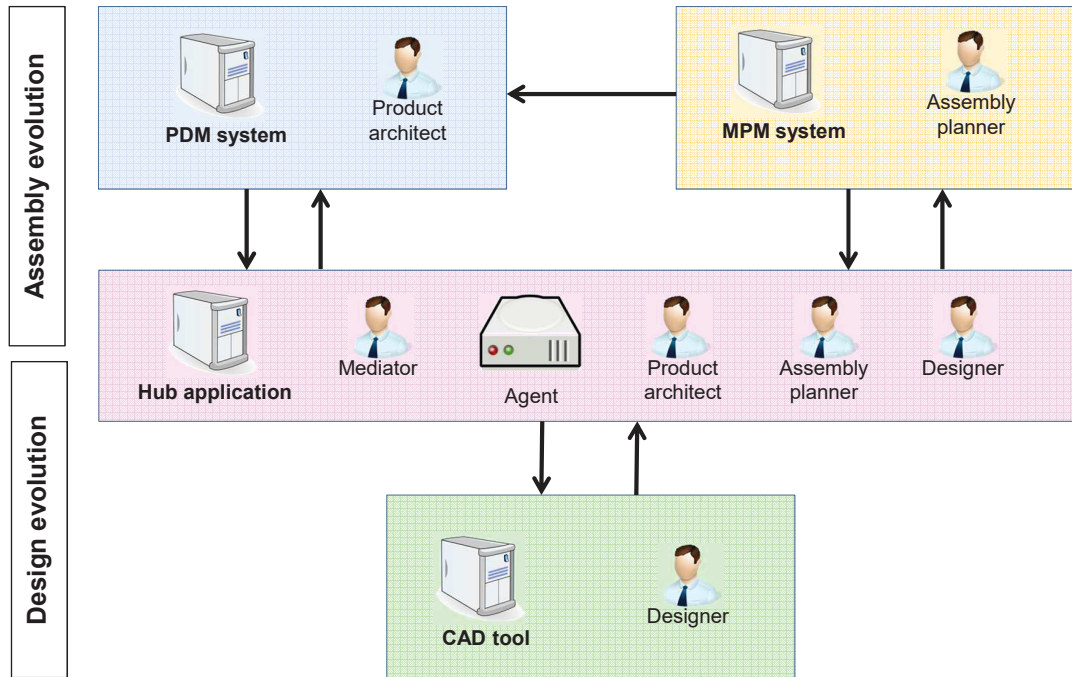


Figure 58: Simplified framework to manage product-process information continuity

The proposed framework is presented in Figure 59 with various mechanisms (grey boxes) and related input/output information (white boxes). The different steps can be deployed as follows:

1. Spatial information (i.e. eBOM, contact relationships and kinematic pairs) and temporal information (i.e. precedence relationships) are queried to the product architect;
2. Swept volumes are generated from previous information;
3. Assembly sequences are automatically generated from the eBOM, contact relationships and precedence relationships. Then, assembly operations are generated from the assembly sequence, as well as the restructured eBOM;
4. The assembly sequence is refined by the assembly planner using temporal mereotopological primitives;

5. The spatiotemporal graph, showing assembly evolution, is generated from the eBOM, mBOM, swept volumes, kinematic pairs and temporal primitives;
6. Assembly and interface skeletons are generated in CAD tools, as well as the product structure (based on the restructured eBOM);
7. The designer creates the parts in a CAD application;
8. The designer is queried if a change is needed for a part. If so, the designed part is changed;
9. The spatiotemporal graph, showing design evolution, is generated from changes listed by the designer along the design process;
10. Product can be visualized.

Here the approach starts at the architectural phase, after having defined functions and preliminary solutions. In the framework, the spatiotemporal graph gives to product architect and designer the opportunity to have an overall view of the product evolution. Indeed, it gathers, represents and links information from PDM and MPM. With such a graph, spatiotemporal reasoning is enabled and stakeholders can understand how entities are linked over space and time, and how they interact. The fact of gathering information on one graph improves the understanding of product architect and designer and they are more aware of design intents [Peachavanish et al., 2006]. The proposed approach does not consider the generation of assembly and interface skeletons, as well as product structure in a CAD application. Indeed, the link with CAD tools will be developed in future works.

Moreover, this approach considers new entities that are not taken into consideration by current information systems. For instance, information (such as spatiotemporal relationships describing the assembly and design evolution), requires to be integrated in an IT manner. But neither PDM nor MPM can manage this kind of information. As such, a lack of interoperability among PLM systems is highlighted. PDM and MPM require a common semantic, which allows having the same vocabulary between stakeholders of a project and provides a support to knowledge management. Moreover, a top-management level should be added so as to manage new entities, objects and their related relationships, and ensure a coordination, as well as an understanding of the different information systems to each stakeholder. Indeed, interoperable tools are required to manage heterogeneous information [Peachavanish et al., 2006]. Besides, some change management can just occur on one way (cf. top of Figure 60): from PDM to MPM systems. On the contrary, no link exists from MPM to PDM. Therefore, the introduced hub will be detailed in the next session in order to avoid heterogeneity issues between information systems and be able to manage multi-dimensional information. Moreover, the hub will enable the information both ways between PDM and MPM (cf. bottom of Figure 60).

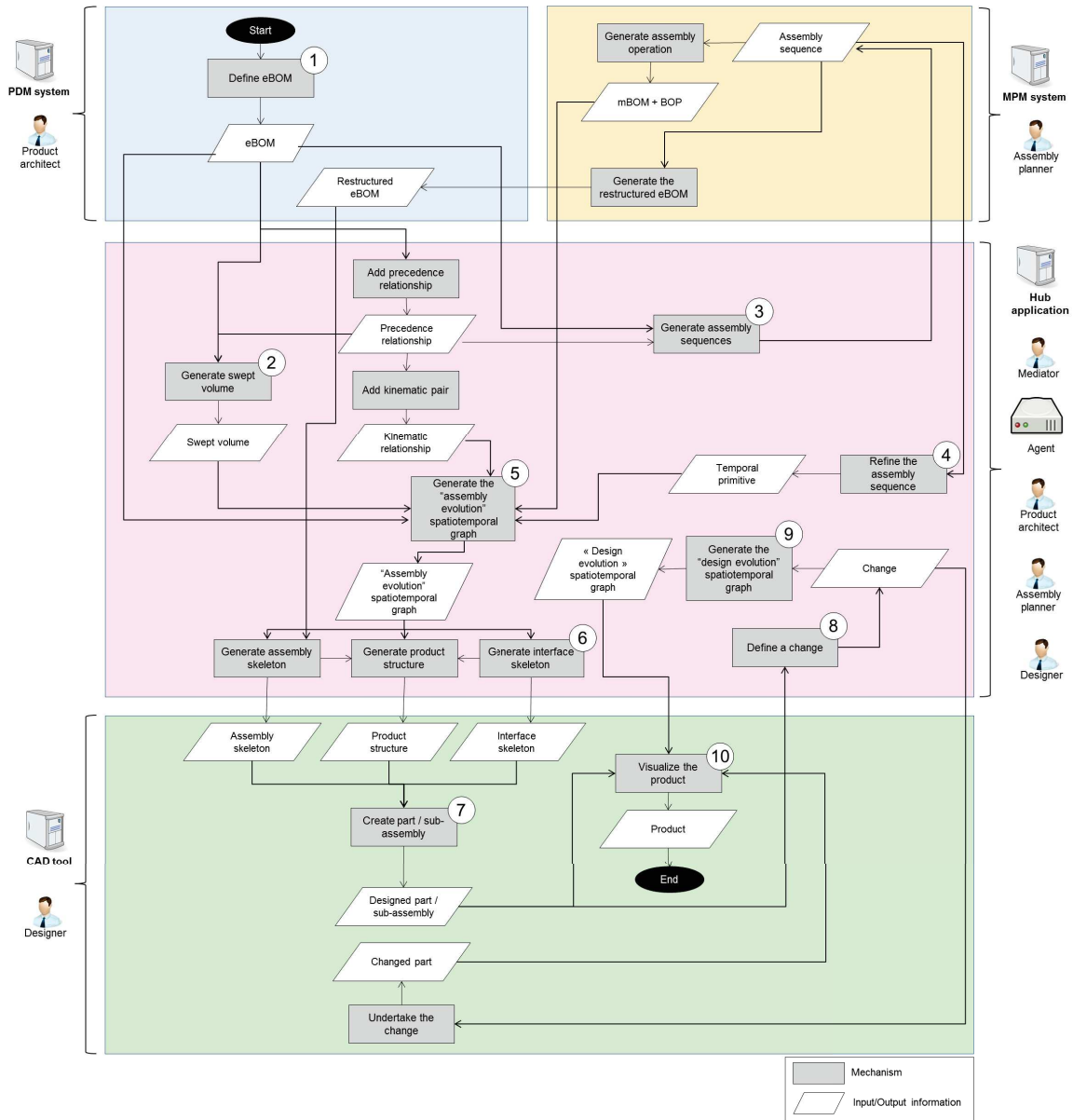


Figure 59: Proposed framework to manage product-process information continuity in the context of AOD

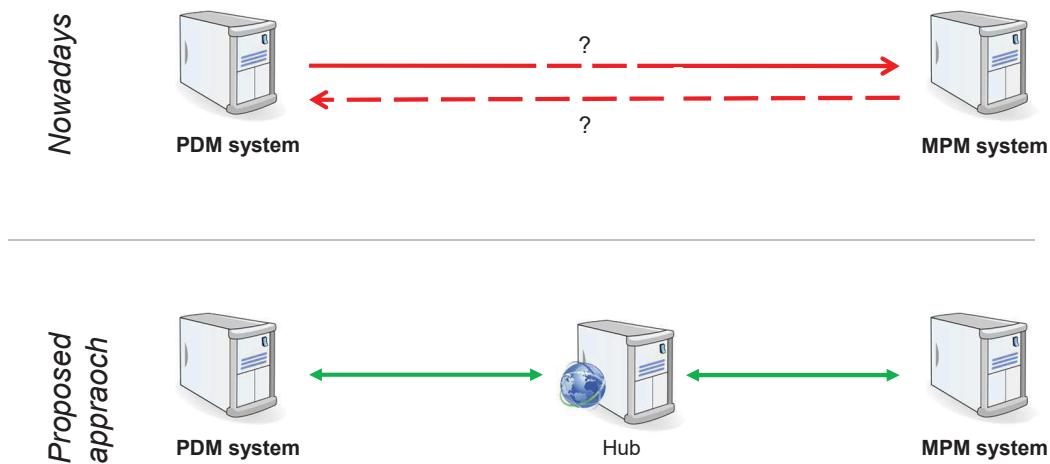


Figure 60: Proposed approach to bridge the gap of heterogeneity

## 5.6 Overview of information flow through PLM

This section introduces the information flow through PLM systems. Figure 61 shows the evolution of information systems implementation. In a concurrent engineering project, information is exchanged among CAD, PDM, MPM systems and so on. In fact, information systems are often related to each other in pairs in order to share information [Ciocoiu et al., 2001]. If the company owns several systems, one system is linked to all systems (left side of Figure 61). The relationships between systems are complex, not well optimized and sometimes just one-way.

As such, the authors propose to link all information systems to the hub (right side of the Figure 61) and to delete links between information systems, so that the gap of heterogeneity is decreased. The hub is linked to PRONOA2 ontology with a double arrow, which ensures the interoperability between systems by providing a common semantic among PDM, MPM and CAD. Semantics allows having the same vocabulary between stakeholders of a project and provides with ontology a support to knowledge management. Besides, double arrows are present at each link. As such, bridge enables transferring information from one information system to another. All information systems have been separated so that they are all independent: if a software editor chooses to make a modification on its own tool, it will not impact the others. In addition, the overall approach enables the semi-automation of predictable design by supporting designers in their decisions. Here the hub is considered as the dashboard, which controls information flow.

The MERCURY approach has been inspired by the musical domain. An orchestra is for instance composed of wind (here PDM system) and brass (here MPM system) instruments. The orchestra is guided by the conductor (here the hub), which gives the rhythm. Each family of instruments (here information systems) includes musicians (here agents).

Each musician will make his own sound (here link between the information systems and hub), which will be heard by the conductor. All sounds, coming from wind and brass instruments, create a sheet music (here the ontology is instanciated). If a music is created by all instruments family, then it is a full symphony.

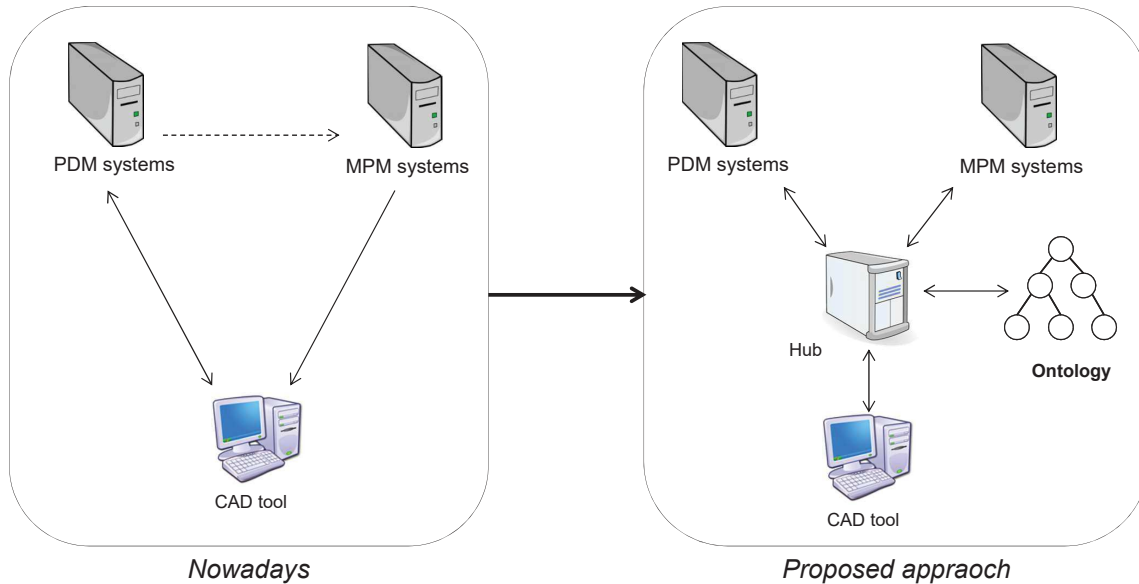


Figure 61: Evolution of PLM systems and CAD application implementation

Moreover this approach, based on relationships management, enables the reconciliation of design activities. Indeed, product evolution is controlled and information is provided to stakeholders as early as possible in the design process. Figure 62 presents the entities, which travel through PLM systems (i.e. PDM and MPM systems) and CAD application. Dotted arrows represent information that enters and full arrows, information given to information systems and applications. The PDM system exports the eBOM filled by the product architect and receives the restructured eBOM based on assembly sequence. The MPM system receives the mBOM after that the assembly sequence has been found by ASDA algorithm and exports the temporal regions and primitives. The CAD application receives the CAD tree and sends back the geometric information (such as geometric skeletons), as well as the changes undertaken by the designer along the design process.

Figure 63 illustrates in details the different steps followed by the information flow at the early design stages through the hub. The chosen hub is Pegasus, developed by Demoly et al. [Demoly et al., 2011a]. Pegasus is a web-based system and a mediator application between information systems (i.e. PDM and MPM) and CAD tools in the AOD [Demoly et al., 2013a]. Its role is to manage product relationships and control information (with spatial, temporal and spatiotemporal dimensions) flow between information systems and ontology with internal regulation procedures, so as to bridge the heterogeneity gap.



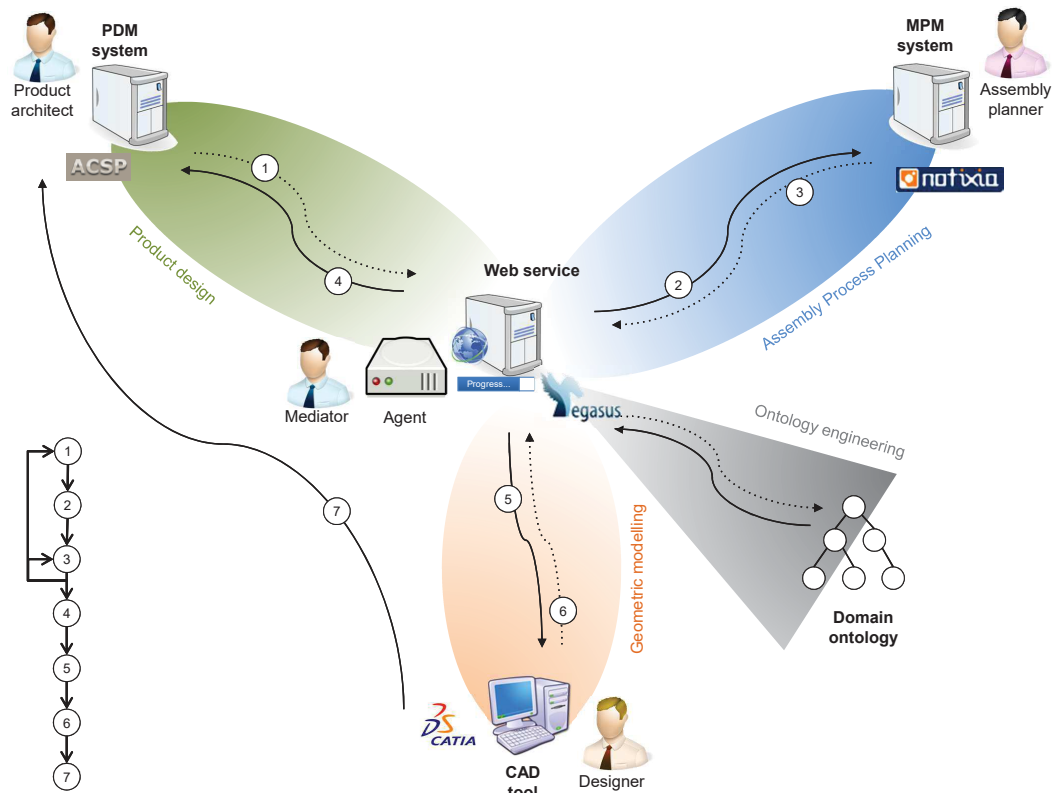


Figure 62: Type of information travelling through PLM systems and CAD application

Moreover, Pegasus is a decision making support for sub-assemblies selection and assembly operation structures generation by capturing DFA rules. So Pegasus' purpose is to bring concurrent engineering benefits into the product design and assembly sequence planning stages. This hub has been improved in order to take into consideration the new functionalities and new entities to manage.

Table 32 specifies the eleven functions that Pegasus must do and the related stakeholder for each function. In Figure 63, double arrows are used as the user can decide to correct or modify the data at any time of the design process and because the information flows both ways. This figure has some common points with Figure 59, which introduces the proposed framework. The different steps of the proposed approach are presenting below:

**Step 1** The product architect manually fills the eBOM in the PDM system (i.e. ACSP). ACSP is a legacy web-based PLM system, used here as a PDM. Then, PRONOIA2 ontology is automatically and directly populated with the spatial information from PDM. Spatial information corresponds with the spatial regions (e.g. part). A web service or **API (Application Programming Interface)** is created to extract relevant information from PDM and MPM and transfer it automatically into ontology to



generate instance-ontology. Information is first transferred to the internal model of Pegasus, as the ontology does not use the same IT language as information system and as they do not deal with the same information (i.e. PDM deals with all information about the product and the ontology just deals with the e-BOM). As such, the information needs to be transformed to be accessible to PRONOIA2 ontology. The product architect also provides the part-to-part relationships graph (e.g. kinematic pairs, technological pairs and contact) in Pegasus. Kinematic pairs and technological pairs are directly implemented into the ontology as spatiotemporal regions.

**Step 2** Based on the part-to-part relationships, the ASDA algorithm computes several possible assembly sequences and the assembly planner chooses one in Pegasus. The mBOM is then automatically exported to the MPM system (i.e. Notixia). Notixia is a commercial platform to manage assembly process information. The different assembly operations are created in the MPM. The assembly planner refines the assembly sequence, which only uses the *Precedence* primitive between two operations, with other temporal primitives.

**Step 3** Information from MPM is exported to Pegasus so as to populate the ontology with temporal regions and primitives. Then Pegasus behaves like a conductor, as it synchronizes information from PDM and MPM. The user must click on a button to activate the synchronization to update the system. As such, the ontology is automatically populated with the missing information (i.e. spatiotemporal regions). All information is now known by the ontology, which can reason and check the information consistency. If an inconsistency appears, the ontology sends a message to the concerned systems (either PDM or MPM) and the related user must do some changes to avoid the inconsistency. The user cannot go further, if the consistency issue is not solved.

**Step 4** When no inconsistency is detected, the eBOM, based on the assembly sequence, is restructured and stored in PDM.

**Step 5** The representation tool is automatically created to support designer's understanding. Then, the CAD tree is automatically built and exported to CAD application (i.e. Catia V5), with the skeleton-based geometry (i.e. composed of assembly and interface skeletons) related to the previous information (i.e. kinematic and technological pairs).

**Step 6** Based on the assembly skeleton structure, the designer can allocate volume, shape and values related to each skeleton. The geometric information is then sent to Pegasus, which will capitalize the design intents. Besides, the designer has the opportunity to undertake changes, which are kept for traceability in Pegasus hub.

**Step 7** Once designer has defined all product geometric characteristics, the CAD models are stored by using check-in/check-out procedures in PDM system.

Table 32: Functions realised in Pegasus and their related stakeholders

Product architect	Use functional requirements
	Define geometric requirements
	Define part-to-part relationships
	Define product structure
	Define assembly and interface skeletons
Assembly planner	Define manufacturing context
	Generate admissible assembly sequences
	Select sub-assemblies
	Evaluate sub-assemblies
	Evaluate admissible assembly sequences
Designer	Define optimal assembly sequence
	Define design changes
Mediator	Advise the stakeholders thanks to the information traceability and make some prediction
Agent	Store the incoming information, which are not managed by PDM, MPM and CAD
	Check the information consistency
	Generate solutions to the queries from stakeholders
	Generate assembly evolution spatiotemporal graph
	Generate design evolution spatiotemporal graph

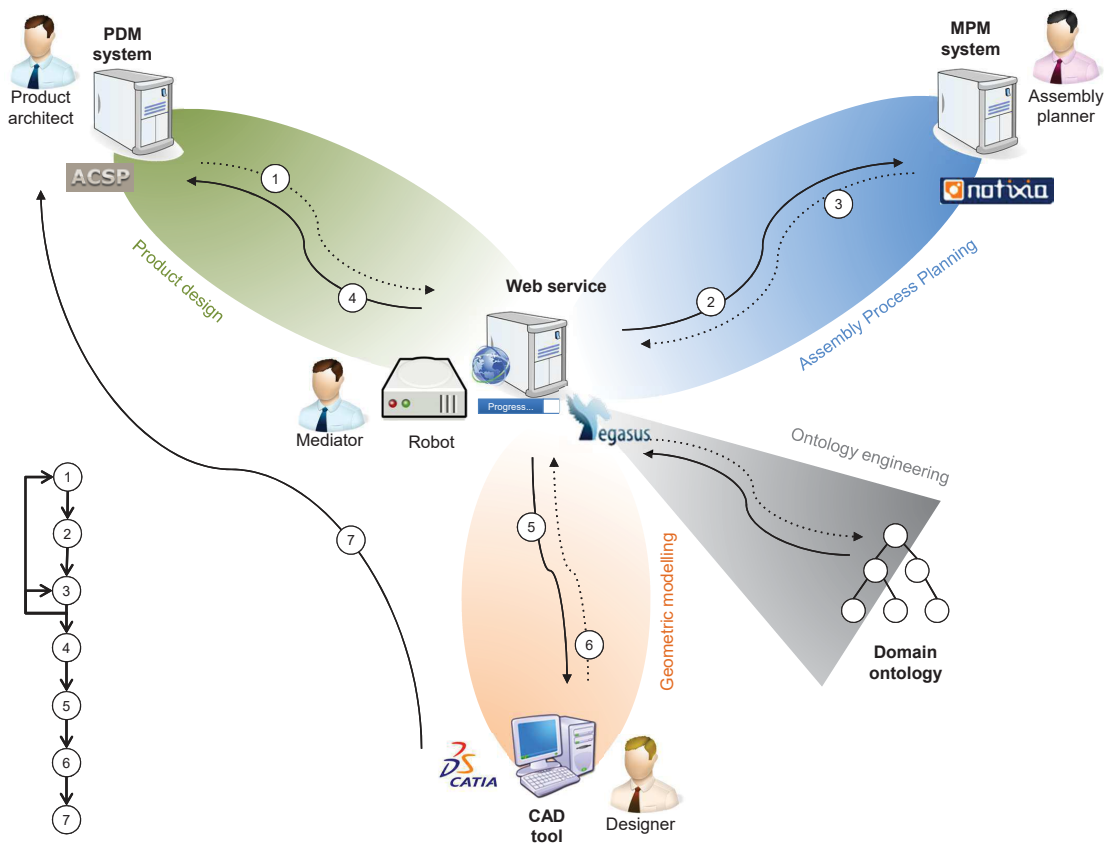


Figure 63: Proposed implementation of PRONOA2 ontology within the hub

If a part is missing or the designer wants to change something, then he can restart the whole process in order to visualize the new results. Moreover, the Pegasus web service has been improved by integrating an ontology and a representation, to become a Semantic Web-service. PRONOA2 ontology is populated at each step. The inference reasoner (here Hermit 1.3.7), which processes checking procedure, gives back information about the consistency. Pegasus, with the aid of the mediator and the agent, has to support three types of stakeholders during the product development process, namely, the product architect, assembly planner and designer. As such, Pegasus and the ontology are a platform for collaborative project requiring the use of PDM, MPM and CAD. They ensure interoperability between information systems and applications and improve communication and information exchange through PLM systems by enhancing visibility with real-time information flow. Therefore, stakeholders can successfully managed the project, as it requires effective control of the exchange of information between stakeholders [Baldwin et al., 1998]. Table 33 sums up which information is brought to designers with the different applications (i.e. PDM, MPM and CAD). Moreover, Table 34 introduces the novel functionalities and concepts handled by the developed hub.

Figure 64 introduces more specifically the information flow (represented by arrows) between PLM systems and PRONOA2 ontology through Pegasus hub for the assembly

Table 33: Information available in the different information systems and applications

PDM	MPM	CAD
eBOM	mBOM	CAD BOM
Subassembly	Operations structure	Part geometry
Part	Assembly operation	Geometric parameter
Product structure	Assembly time	Form feature
Document	Document	Assembly constraints
Configuration		Subassembly geometry

Table 34: Functionalities and concepts of the hub

	Functionalities	Concepts
Information	Management of contact relationship	Contact relationship
	Management of precedence relationship	Precedence relationship
	Management of kinematic pair	Kinematic pair
	Management of technological pair	Technological pair
	Management of design changes relationship	Design changes relationship
	Management of temporal primitive	Temporal primitive
	Management of swept volume	Swept volume
Transferred information	Management of eBOM	eBOM
	Management of spatiotemporal primitive	Spatiotemporal primitive
	Management of assembly operation	Assembly operation

evolution application. The ontology and Pegasus are divided into three areas related to the three dimensions: spatial, temporal and spatiotemporal. Each area is directed by an agent. The role of an agent is to detect an event (e.g. creation of a mechanical part in PDM). Event can be detected with action logics, which describe states that are traversed by actions. Action logics are formal systems that address the activities undertaken by reasoning agents [Salustri, 2012]. As such, design processes are implicitly represented [Salustri, 2000]. Then, the agent interprets this information and builds a XML file, which travels from information systems to Pegasus hub through web service, when the user merges the information. Finally the agent instanciates the instance-ontology. The information coming from PDM is stored in the spatial part of Pegasus and the information coming from MPM is stored in the temporal part. The user has the power to choose when the information is merged and sent to Pegasus. At each merge, the information consistency is checked with the reasoner contained in the ontology. Here are the different steps describing the information flow between PLM systems and Pegasus hub from a computer

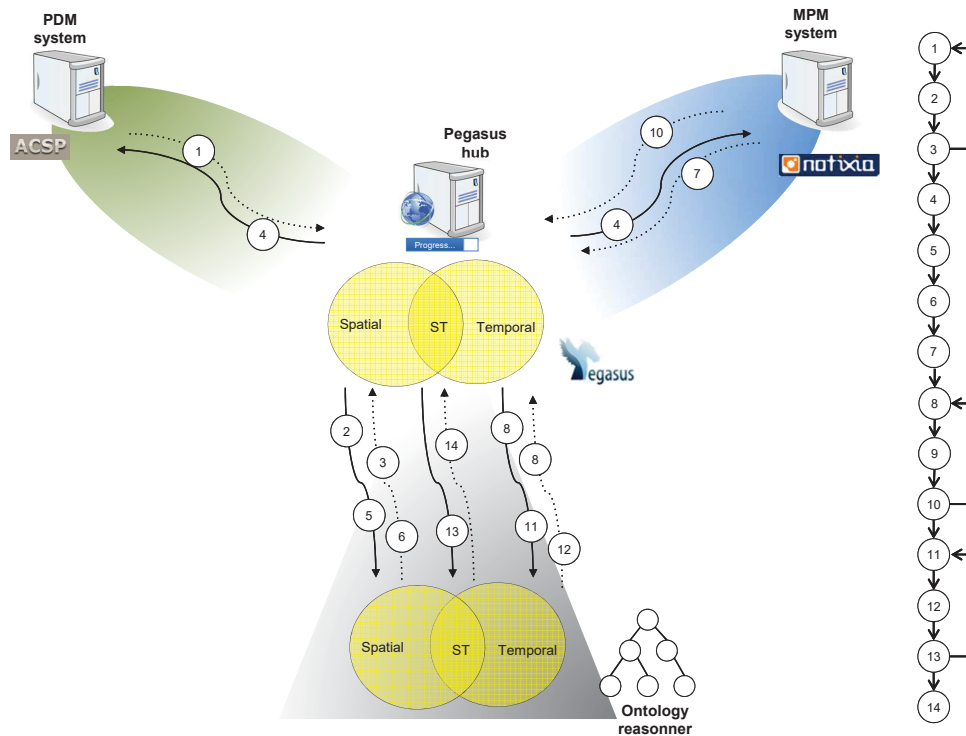


Figure 64: Information flow centered on PRONOIA2 ontology for the assembly evolution application

point of view:

1. Transfer of spatial information from PDM to the spatial part of Pegasus;
2. Transfer of spatial information from Pegasus to the spatial part of PRONOIA2;
3. Check of the information consistency;
4. Transfer information from Pegasus to PDM (and automatically create sub-assemblies in the eBOM) and MPM after that the product architect has given the contact and precedence relations in Pegasus and has validated the assembly sequence proposed by ASDA;
5. Transfer information on sub-assemblies from the spatial part of Pegasus to the spatial part of PRONOIA2;
6. Check of the information consistency;
7. Transfer temporal information (i.e. temporal region) from MPM to the temporal part of Pegasus;

8. Transfer temporal information from the temporal part of Pegasus to the temporal part of PRONOIA2;
9. Check of the information consistency;
10. Transfer temporal information (i.e. temporal primitive) from MPM to the temporal part of Pegasus;
11. Transfer temporal information from the temporal part of Pegasus to the temporal part of PRONOIA2;
12. Check of the information consistency;
13. Transfer spatiotemporal information from the spatiotemporal part of Pegasus to the spatiotemporal part of PRONOIA2 after that the product architect has given the appropriate kinematic pairs between parts in Pegasus and that the algorithm has computed the swept volume;
14. Check of the information consistency.

Moreover, the right side of Figure 64 presents the chronological steps. In fact, stakeholders must follow this chronology and cannot begin to define temporal information before defining spatial information. Flashback are possible, when the reasoner detects inconsistency.

Figure 65 illustrates how information is managed through product development process. Information is first imported from PDM and MPM systems. This information populates the PRONOIA2 ontology and is then stored in the Pegasus hub. The storage depends on the dimension of the information. Indeed, spatial information cannot be stored in the temporal module of Pegasus, thanks to DL rules imposed in the ontology. Finally, the information is displayed in a spatiotemporal graph, so as to provide designers an overview of the product assembly over time.

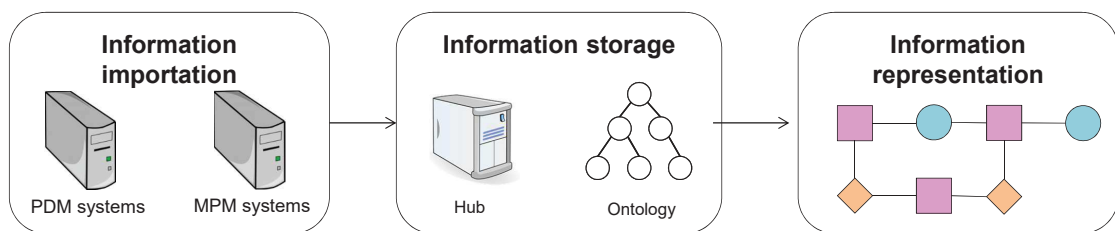


Figure 65: Information progress through the product development process

The MERCURY approach enables having consistent information from PDM and MPM. As such, a feasible assembly can be designed by designer. The consistency is checked in PRONOIA2 reasoner, implemented in Pegasus. The approach also considers a novel

dimension: the spatiotemporal dimension, which enables the description of product evolution over time with spatiotemporal relationships.

## **5.7 Summary**

This chapter has introduced a novel approach, called MERCURY, based on relationships management to describe product evolution through PLM. The approach is based on previous chapters 3 (i.e. JANUS theory) and 4 (PRONOIA2 ontology). Here PRONOIA2 ontology has been implemented in a hub called Pegasus, so as to bring a semantic and logic layer to PLM procedures.

Besides, Pegasus has been improved so as to support assembly and design evolutions and the spatiotemporal dimensions (especially swept volume and spatiotemporal relationships). Indeed, products dynamically evolve through design and assembly phases. Spatiotemporal entities enable orchestrating information from PDM, MPM and CAD systems and control the flow. Moreover, two spatiotemporal graphs have been developed in Pegasus so as to provide a whole overview of product evolution and changes throughout its lifecycle. With such graphs, design intents are communicated to stakeholders and their understanding improves. As such, Pegasus, with the mediator and the agent, keeps information traceability throughout the design process, which could be useful for future designs. Therefore, the proposed approach enables designing consistent product over space and time in a concurrent and proactive context.

# Chapter 6

## Case studies illustrating the theory, model and approach

### Contents

---

<b>6.1</b>	<b>Introduction</b>	<b>124</b>
<b>6.2</b>	<b>First case study: Simplified academic case study</b>	<b>124</b>
6.2.1	Mereotopological description of the case study	124
6.2.2	Swept volumes representation	125
<b>6.3</b>	<b>Second case study: Design of a medical drone</b>	<b>129</b>
6.3.1	Description of the case study	129
6.3.2	Case study solving following the approach on assembly evolution	129
6.3.3	Populating PRONOIA2	134
6.3.4	Reasoning - consistency checking	135
6.3.5	Case study solving following the approach on design evolution	136
<b>6.4</b>	<b>Summary</b>	<b>142</b>

---



## 6.1 Introduction

This section introduces two mechanical case studies with their related mereotopological descriptions, application-ontologies and representations to improve product architect's and designer's understanding.

## 6.2 First case study: Simplified academic case study

This case study focuses on the mereotopological description of the assembly and on the novel entities called swept volumes.

### 6.2.1 Mereotopological description of the case study

For this first case study, the main function is to stabilize the hub on the axis. A solution is to use a nut to grip the hub in translation. The search of pre-concept is assumed to have already been carried out. Here, the product is composed of six parts linked together with kinematic pairs (as represented in Figure 66).

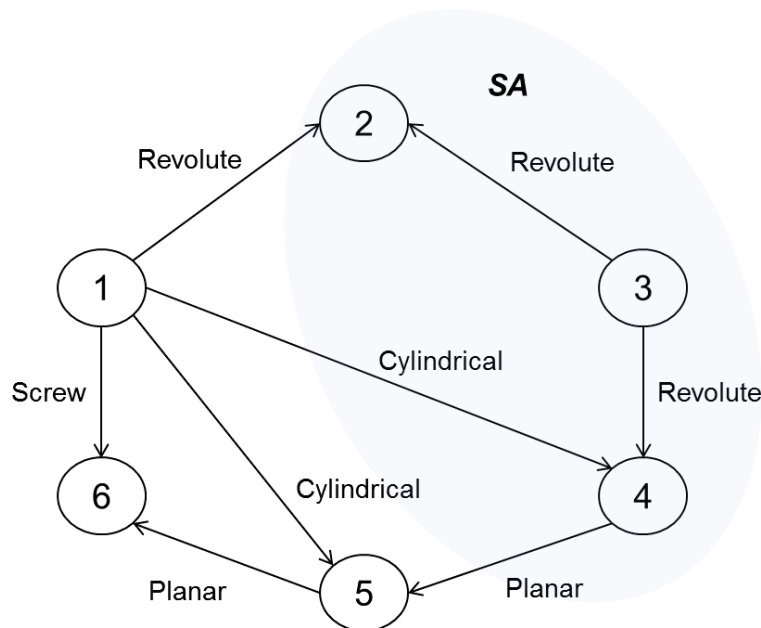


Figure 66: Graph of part-to-part relationships of the first case study

After generating the assembly sequence (with the ASDA algorithm), every kinematic pairs can be translated into spatiotemporal mereotopological primitives related to a specific assembly operations (cf. Table 35). Here *Parts 2, 3 and 4* compose the Sub-Assembly (SA). Each primitive is defined from Tables 13–14 and Tables 20–23 and adapted to the case study.

A spatiotemporal graph (cf. Figure 67) is also generated in the hub so as to have a complete overview of the relationships between regions at the different assembly operations. This graph enables to sum up all previous spatial, temporal and spatiotemporal information.

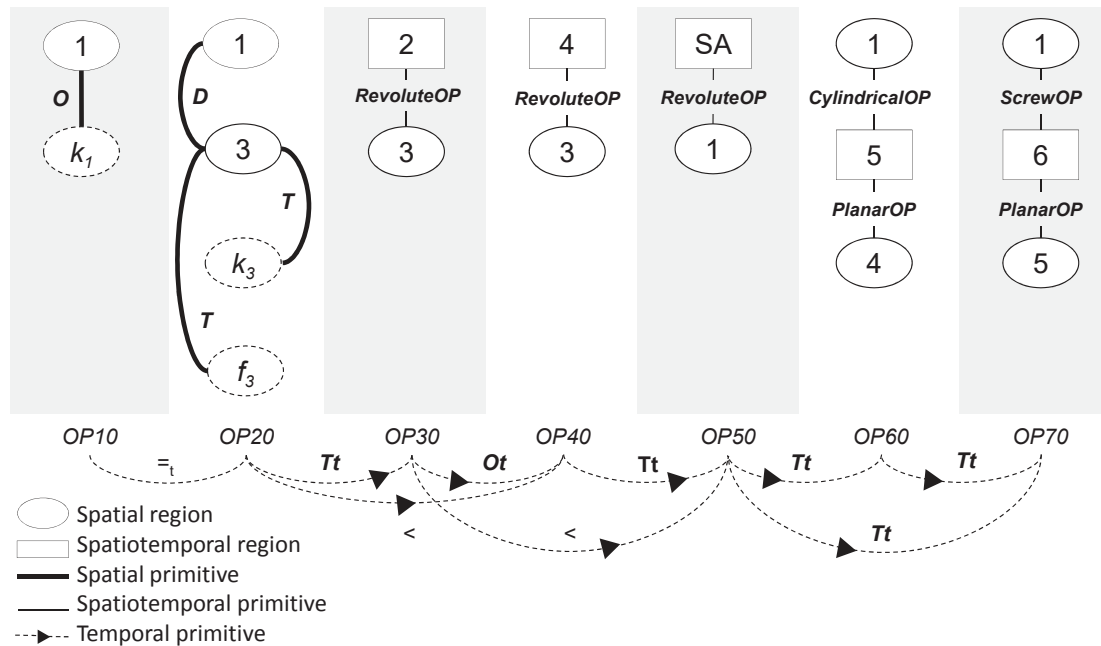


Figure 67: Description of the whole assembly process in a spatiotemporal graph for designers – first case study

A purely spatial graph representing the final state of the assembly process is introduced in Figure 68. This figure represents the spatial relationships between parts or sub-assemblies.

### 6.2.2 Swept volumes representation

After well understanding the product architect's intents, the designer can design the assembly using a CAD tool (i.e. Catia v5). The overall assembly is represented in Figure 69 and the eBOM is defined in Table 36

Table 36: Parts list of the first case study

No.	Part name	No.	Part name
1	Shaft	4	Bearing 2
2	Bearing 1	5	Spacer
3	Hub	6	Nut

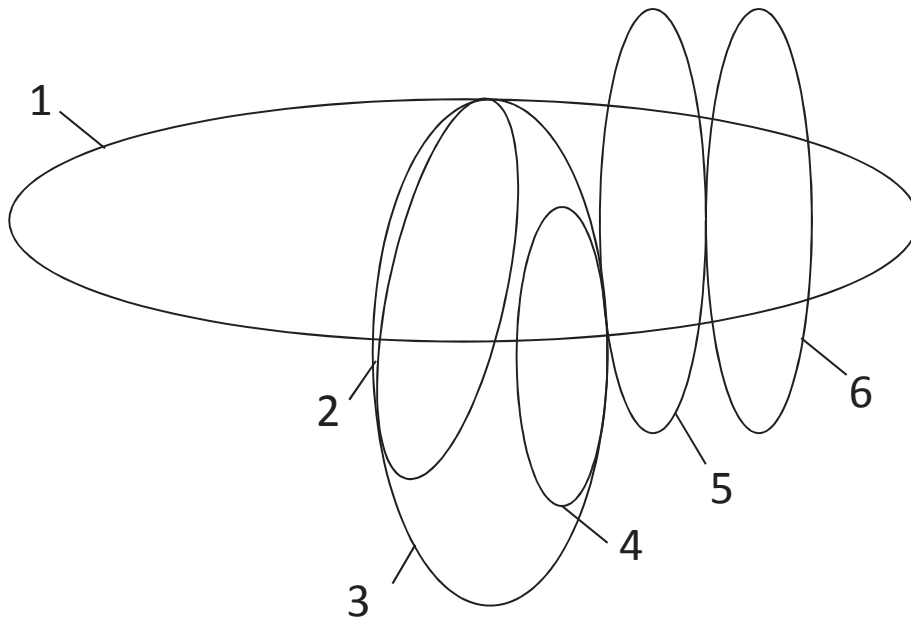


Figure 68: Spatial representation of the whole assembly process for the first case study

Moreover, designers can check how parts are assembled over time, their path to reach their final destination and the space needed during the move. As such, Figure 70 presents the design evolution for particular assembly operation. For instance, at the *OP60* assembly operation, *Part 1* and *Part 4* are in contact with *Swept Volume 5* (*Part 5* is currently moving to reach its final position). The spatiotemporal region (i.e. Swept Volume), which is visible in the Figure 70, aids the designer to understand the part evolution (invisible with human eyes) and the path that part follows during the assembly process. The last picture of Figure 70 presents only the overall assembly with swept volumes.

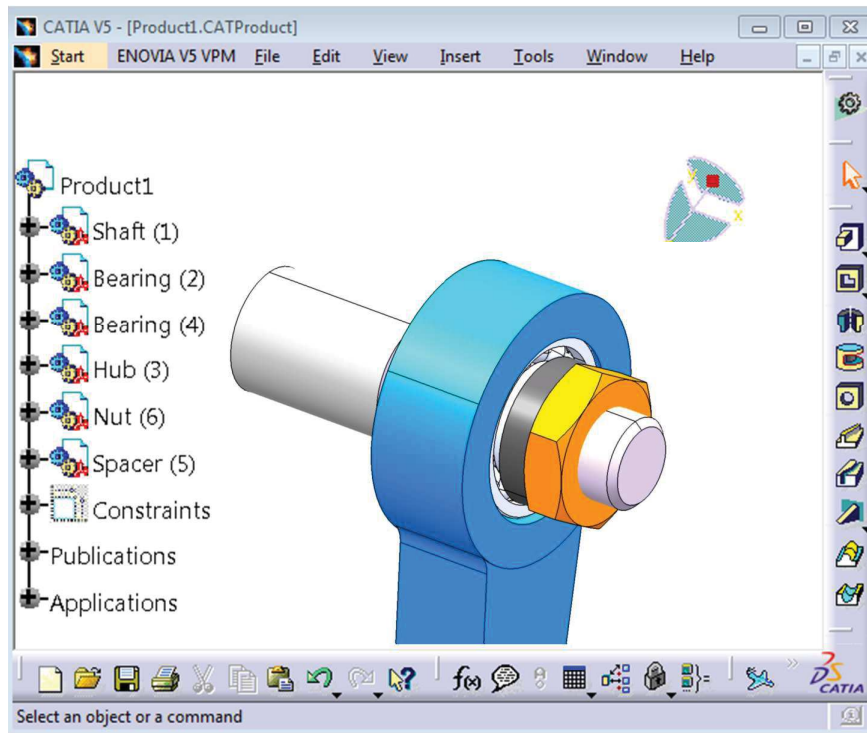


Figure 69: CAD representation of the whole assembly within CATIA v5 for the first case study

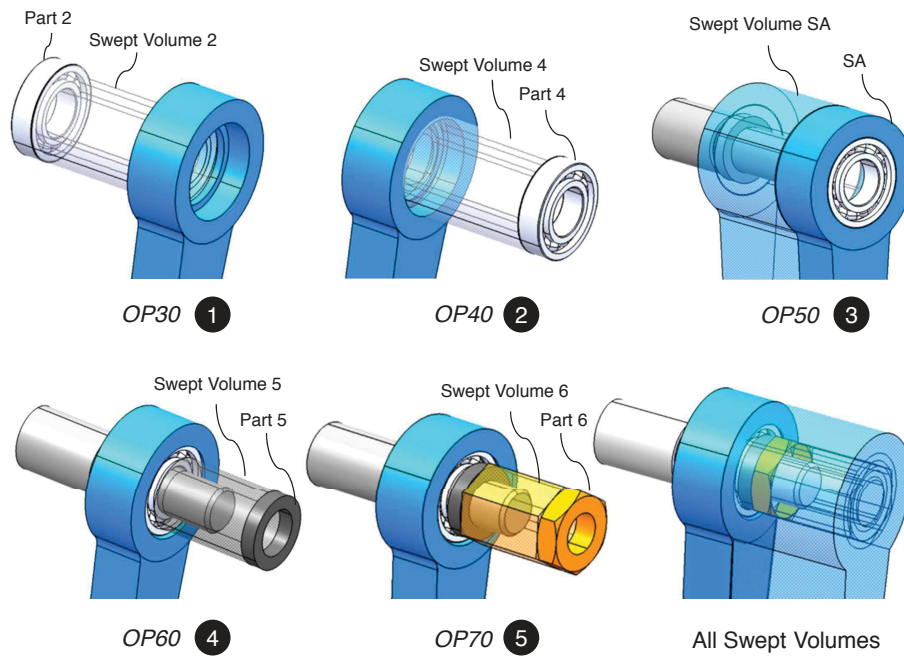


Figure 70: Design evolution of the first case study step by step

Table 35: Mereotopological description of the first mechanical case study

OP No.	Spatiotemporal primitive	Mereotopological description	Skeleton
10	2 <b>RevoluteOP</b> 3	$((3\mathbf{D}2) \wedge (3\mathbf{O}k_1) \wedge (2\mathbf{D}k_1))_{TR1} \wedge (2\mathbf{Move}3)_{TR2} \wedge$ $((3\mathbf{D}2) \wedge (3\mathbf{O}k_1) \wedge (2\mathbf{O}k_1))_{TR3} \wedge (2\mathbf{Move}3)_{TR4} \wedge$ $((3\mathbf{X}2) \wedge (3\mathbf{O}k_1) \wedge (2\mathbf{O}k_1))_{TR5} \wedge (2\mathbf{Move}3)_{TR6} \wedge$ $((3\mathbf{T}2) \wedge (3\mathbf{X}2) \wedge (3\mathbf{O}k_1) \wedge (3\mathbf{T}k_2) \wedge (3\mathbf{T}f_2) \wedge$ $(2\mathbf{O}k_1) \wedge (2\mathbf{T}k_2) \wedge (2\mathbf{T}f_2))_{TR7}$	$k_1$ line, $k_2$ plane, $f_2$ surface
20	4 <b>RevoluteOP</b> 3	$((3\mathbf{D}4) \wedge (3\mathbf{O}k_1) \wedge (4\mathbf{D}k_1))_{TR1} \wedge (4\mathbf{Move}3)_{TR2} \wedge$ $((3\mathbf{D}4) \wedge (3\mathbf{O}k_1) \wedge (4\mathbf{O}k_1))_{TR3} \wedge (4\mathbf{Move}3)_{TR4} \wedge$ $((3\mathbf{X}4) \wedge (3\mathbf{O}k_1) \wedge (4\mathbf{O}k_1))_{TR5} \wedge (4\mathbf{Move}3)_{TR6} \wedge$ $((3\mathbf{T}4) \wedge (3\mathbf{X}4) \wedge (3\mathbf{O}k_1) \wedge (3\mathbf{T}k_4) \wedge (3\mathbf{T}f_4) \wedge$ $(4\mathbf{O}k_1) \wedge (4\mathbf{T}k_4) \wedge (4\mathbf{T}f_4))_{TR7}$	$k_1$ line, $k_4$ plane, $f_4$ surface
30	SA <b>RevoluteOP</b> 1	$((1\mathbf{D}SA) \wedge (1\mathbf{O}k_1) \wedge (SA\mathbf{D}k_1))_{TR1} \wedge$ $(SA\mathbf{Move}1)_{TR2} \wedge ((1\mathbf{D}SA) \wedge (1\mathbf{O}k_1) \wedge$ $(SA\mathbf{O}k_1))_{TR3} \wedge (SA\mathbf{Move}1)_{TR4} \wedge ((1\mathbf{X}SA) \wedge$ $(1\mathbf{O}k_1) \wedge (SA\mathbf{O}k_1))_{TR5} \wedge (SA\mathbf{Move}1)_{TR6} \wedge$ $((1\mathbf{T}SA) \wedge (1\mathbf{X}SA) \wedge (1\mathbf{O}k_1) \wedge (1\mathbf{T}k_3) \wedge (1\mathbf{T}f_3) \wedge$ $(SA\mathbf{O}k_1) \wedge (4\mathbf{T}k_3) \wedge (4\mathbf{T}f_3))_{TR7}$	$k_1$ line, $k_3$ plane, $f_3$ surface
40	5 <b>CylindricalOP</b> 1 and 5 5 <b>PlanarOP</b> 4	$((1\mathbf{D}5) \wedge (1\mathbf{O}k_1) \wedge (5\mathbf{D}k_1))_{TR1} \wedge (5\mathbf{Move}x)_{TR2} \wedge$ $((1\mathbf{D}5) \wedge (1\mathbf{O}k_1) \wedge (5\mathbf{O}k_1))_{TR3} \wedge (5\mathbf{Move}1)_{TR4} \wedge$ $((1\mathbf{X}5) \wedge (1\mathbf{O}k_1) \wedge (5\mathbf{O}k_1))_{TR5} \wedge (5\mathbf{Move}4)_{TR6} \wedge$ $((4\mathbf{T}5) \wedge (4\mathbf{T}f_5) \wedge (5\mathbf{T}k_5) \wedge (k_5\mathbf{T}f_5))_{TR7}$	$k_1$ line, $k_5$ plane, $f_5$ surface
50	6 <b>ScrewOP</b> 1 and 6 <b>PlanarOP</b> 5	$((1\mathbf{D}6) \wedge (1\mathbf{O}k_1) \wedge (6\mathbf{D}k_1))_{TR1} \wedge (6\mathbf{Move}1)_{TR2} \wedge$ $((1\mathbf{D}6) \wedge (1\mathbf{O}k_1) \wedge (6\mathbf{O}k_1))_{TR3} \wedge (6\mathbf{Move}1)_{TR4} \wedge$ $((1\mathbf{X}6) \wedge (1\mathbf{O}k_1) \wedge (6\mathbf{O}k_1))_{TR5} \wedge (6\mathbf{Move}5)_{TR6} \wedge$ $((5\mathbf{T}6) \wedge (5\mathbf{T}f_6) \wedge (6\mathbf{T}k_6) \wedge (f_6\mathbf{T}k_6))_{TR7}$	$k_1$ line, $k_6$ plane, $f_6$ surface

## 6.3 Second case study: Design of a medical drone

### 6.3.1 Description of the case study

This case study is based on a project made by students, who were asked to design a medical drone. Medical drone is mainly used to reduce the delay between the alert time and the intervention time. This time can actually be crucial for heart attacks, research of victims and so on. This drone has been thinking as a support for doctors and is complementary to ambulances. This drone has been developed to cover several different intervention situations, as well as to be easy to drive and use.

### 6.3.2 Case study solving following the approach on assembly evolution

The case study is solved by following the different steps of the proposed approach (cf. Figure 59). The objective is to design a medical drone. First, functional analysis has been carried out in order to understand the needs and find some preliminary solutions (cf. Table 37). After defining the requirements, some pre-concepts have been drawn. Based on morphogenesis, one solution is illustrated in Figure 71. Here, the case study has been simplified and only described a part of the whole assembly. The case study border has been marked out in Figure Figure 71 with a black circle.

Table 37: Functions and related solutions for the second case study

Functions	Solutions
Fly in order to bring medical equipment	Four drive turbines
Land on every kind of grounds	Landing gear with three legs
Go anywhere	Retractable rear arms

**Define eBOM** The eBOM is composed of ten parts and is represented in Figure 72 within Notixia.

**Add precedence relationships and kinematic pairs** The part-to-part relationships are defined in a directed graph (cf. Figure 73) by the product architect. In this graph, the contact (i.e. link between two circles), precedence relationships (i.e. arrow) and kinematic pairs between two parts can be visualized.

**Generate swept volumes** Each swept volume is deduced from the part-to-part relationships graph. In fact, swept volumes are presented as the parts touching the end of the arrow. Here the swept volumes are *Swept volume 2*, *Swept volume 4*, *Swept volume 7*, *Swept volume 9*, *Swept volume 10*, *Swept volume SA1*, *Swept volume SA3* and *Swept volume SA4*.

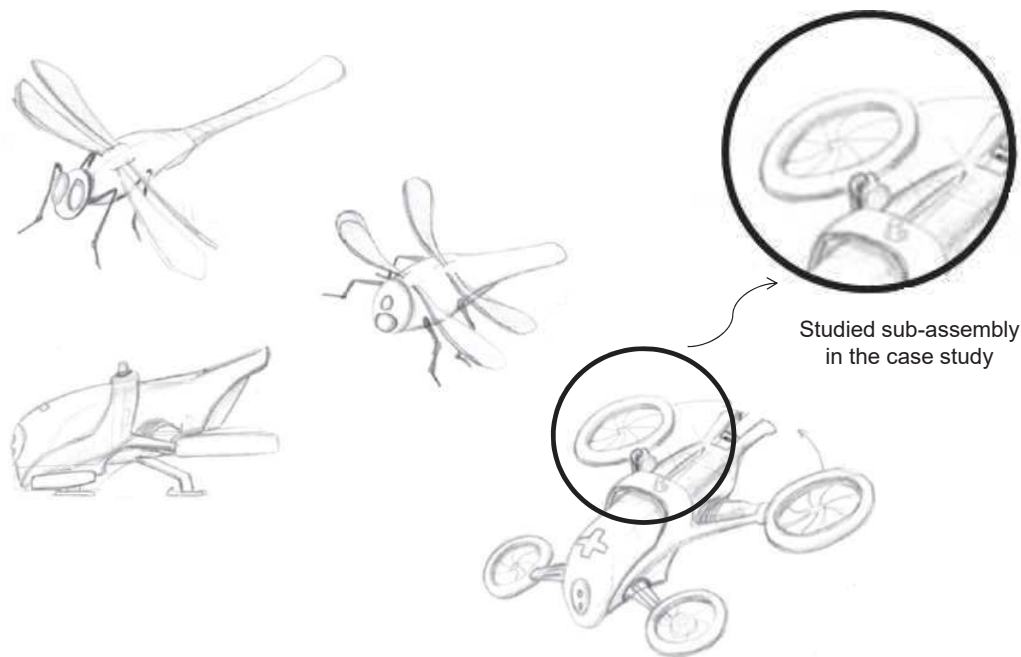


Figure 71: Representation of a pre-concept using morphogenesis for the case study “Medical drone”

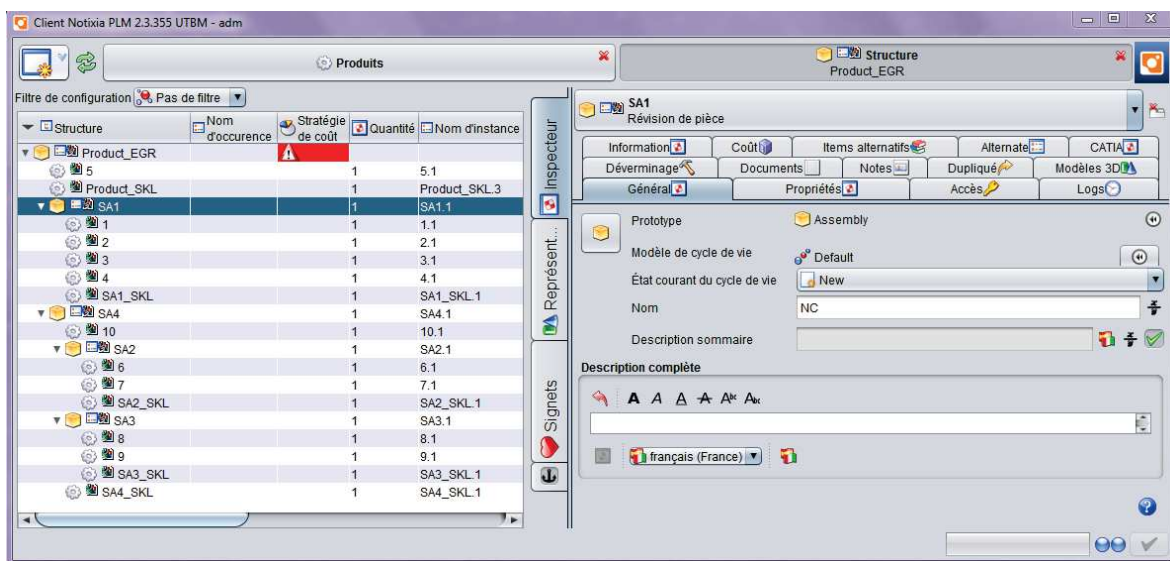


Figure 72: eBOM structure within MPM system (Notixia) for the “medical drone” case study

**Generate assembly sequences, choose the most suitable one and refine it** Based on part-to-part relationships, the ASDA algorithm generates a relevant assembly sequence that is expressed as:



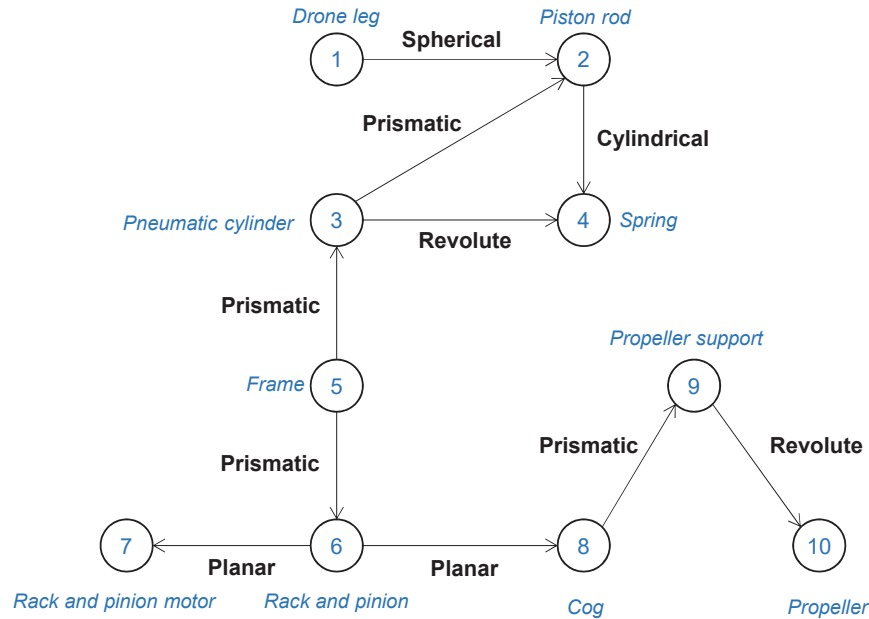


Figure 73: Graph of part-to-part relationships of the second case study

Assembly = [SA1 ; 5 ; SA4]

Assembly = [[1 ; 2 ; 3 ; 4] ; 5 ; [SA3 ; SA2 ; 10]]

Assembly = [[1 ; 2 ; 3 ; 4] ; 5 ; [[8 ; 9] ; [6 ; 7] ; 10]]

So the spatial regions 1, 2, 3 and 4 are embedded in a Sub-Assembly (denoted SA1), which represents the landing system. Besides, the spatial regions 6 and 7 are embedded in a Sub-Assembly (denoted SA2) to represent the rack and pinion system and the spatial regions 8 and 9 are embedded in a Sub-Assembly (denoted SA3) for the arm system. Finally, the spatial regions SA3, SA2 and 10 are embedded in a Sub-Assembly (denoted SA4).

**Generate assembly operations** Operations are defined from the assembly sequence. Here, the different assembly operations are expressed in Table 38 and are represented in Figure 74 within Notixia. The assembly sequence is then refined by the assembly planner and is represented in Figure 75.

**Generate restructured eBOM** The restructured eBOM (cf. Figure 76) considers the parts, as well as the sub-assemblies previously detected. Assembly and interface skeletons are also taken into consideration in the CAD tree.

**Generate the spatiotemporal graph** In the previous directed graph, the temporal information (cf. Table 38) and the kinematic pairs positioning can be converted into mereotopological descriptions with the JANUS theory (cf. Table 39 and Table 40). All information is then gathered to describe product evolution (cf. Figure 77). Hence, spatial (i.e. mechanical part), temporal (i.e. assembly operation) and spa-



Table 38: Temporal information for the second case study: assembly operations and their associated primitives

Operations description [Boothroyd et al., 2002]	Temporal relationships between operations
$OP_{10}$ : Insertion of 4 on 2	$OP_{10} \mathbf{Tt} OP_{20} \wedge OP_{10} < OP_{30} \wedge OP_{10} \mathbf{Pts} OP_{40} \wedge OP_{10} \mathbf{Pts} OP_{50}$
$OP_{20}$ : Insertion of 2 and 4 on 3	$OP_{20} \mathbf{Ot} OP_{30}$
$OP_{30}$ : Insertion of 2 on 1	$OP_{30} \mathbf{Tt} OP_{90}$
$OP_{40}$ : Insertion of 7 on 6	$OP_{40} \mathbf{Tt} OP_{70}$
$OP_{50}$ : Insertion of 9 on 8	$OP_{50} \mathbf{Ot} OP_{60}$
$OP_{60}$ : Insertion of 10 on 9	$OP_{60} \mathbf{Tt} OP_{70}$
$OP_{70}$ : Insertion of SA3 on SA2	$OP_{70} \mathbf{Tt} OP_{80}$
$OP_{80}$ : Insertion of SA4 on 5	$OP_{80} =\mathbf{t} OP_{90}$
$OP_{90}$ : Insertion of SA1 on 5	

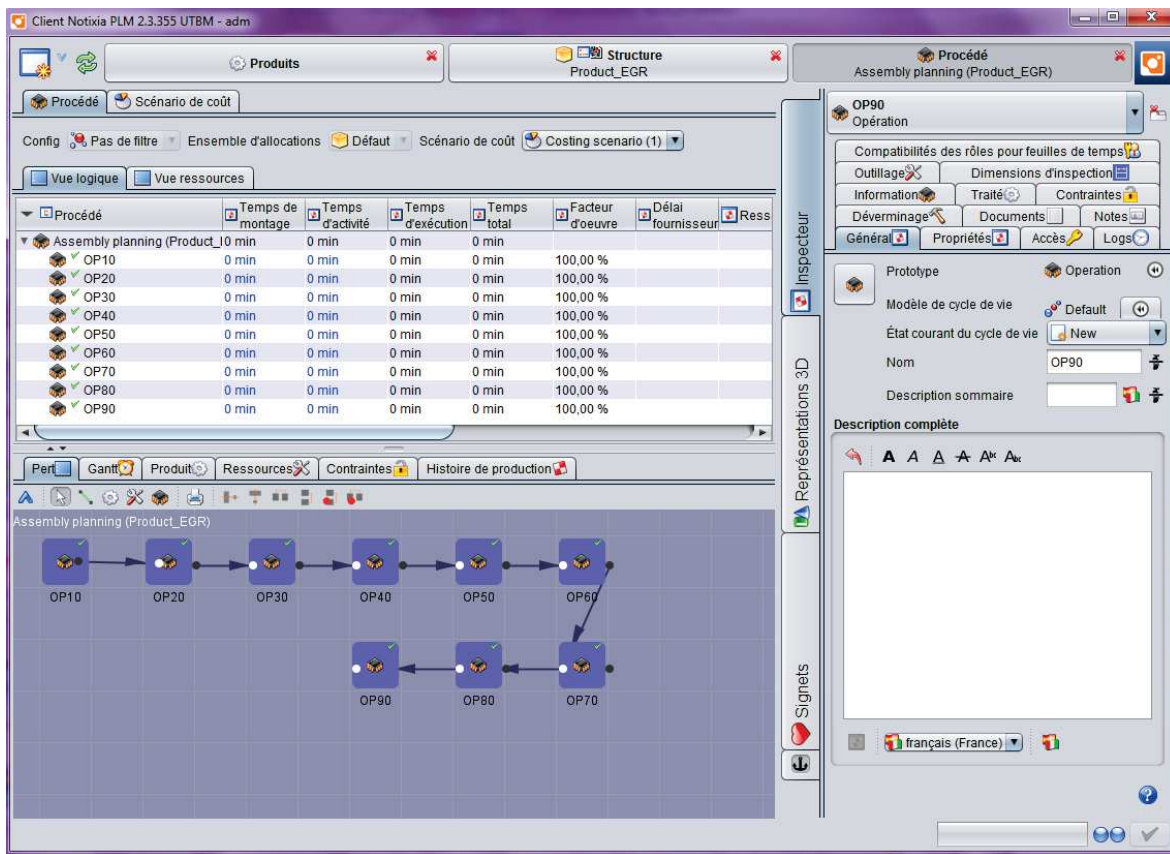


Figure 74: Assembly operations representation within MPM system (Notixia) for the “medical drone” case study

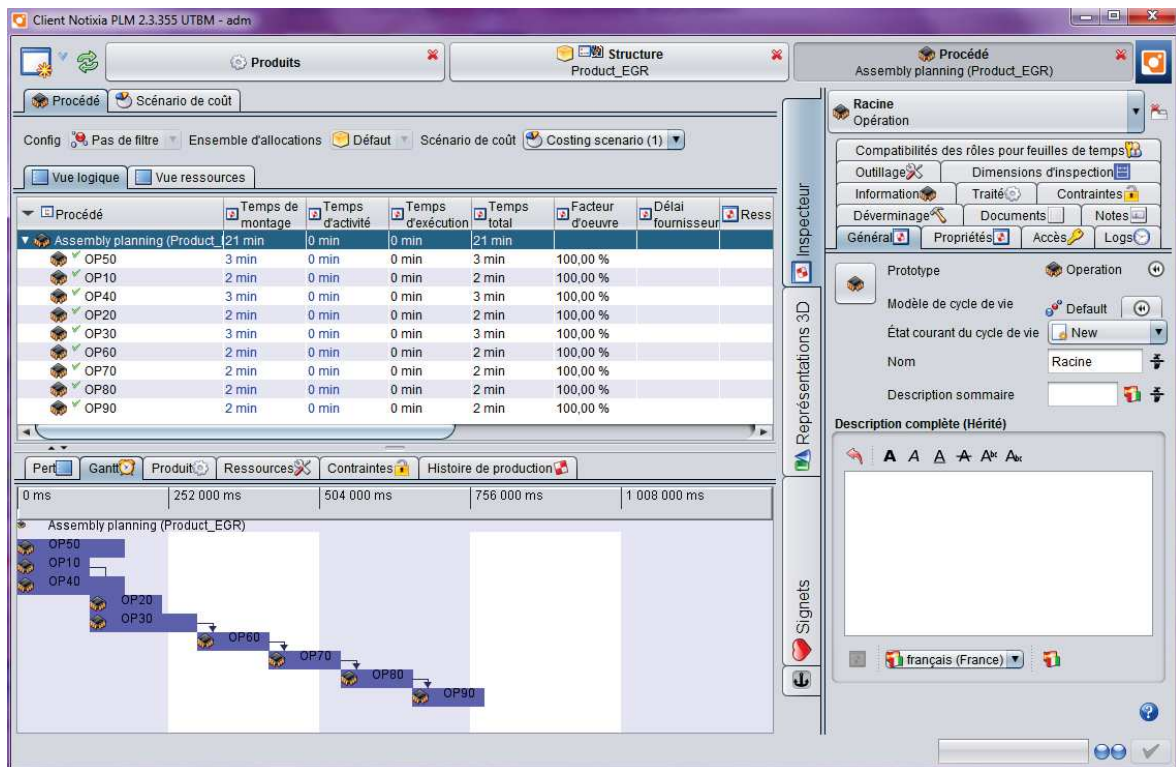


Figure 75: Representation of the assembly sequence planning of the second case study within MPM system (Notixia)

tiotemporal (i.e. swept volume in the context of AOD) regions are represented and linked with primitives. Only regions, which undergo a change at  $OP_i$ , are presented at  $OP_i$  in order to facilitate understanding of the reader. Here, changes are considered when novel spatiotemporal relationships appear. As such, the product architect and designer can understand what happens for each operation, especially where a part is situated and linked regarding to others, and which part is moving or evolving and so on. For instance, he can read that at  $OP_{30}$ , *Swept volume 2* is moving to be positioned in a **SphericalOP** manner on *Part 1*. Then this representation provides an overview of the product evolution along its assembly sequence to the product architect and designer before starting the product modeling phase in CAD. This representation therefore introduces a novel working support to the product architect and designer.

Besides, a purely spatial representation of the final stage of the assembly design process is represented in Figure 78. Kinematic pairs are expressed with mereotopological primitives. For instance **PlanarOP** is equivalent to **Tangent** primitive at the end of the process, as well as **RevoluteOP** is equivalent to **Cross** and **Tangent** primitives. Since JANUS theory is used at the early design stages, qualitative information is favored (detailed geometry does not yet exist). This kind of representation allows the designer to fully understand the relationships between parts at

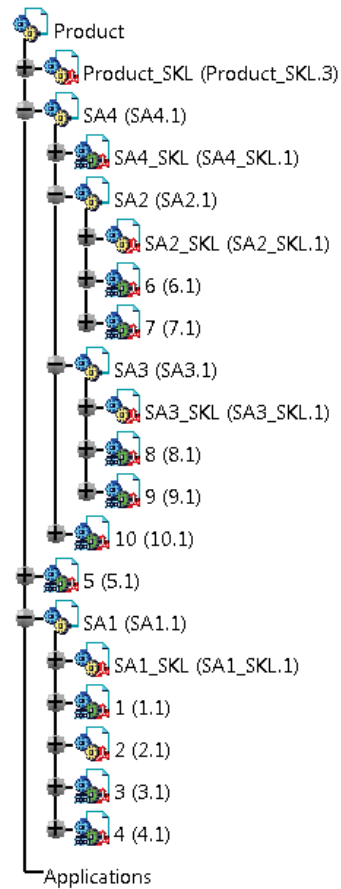


Figure 76: Restructured eBOM of the second case study within CAD application (Catia)

the end of the assembly process. For example, such graph informs that *Part 10* and *Part 5* are never in contact and that some parts can be gathered in a sub-assembly. This information will aid the designer in its CAD modeling activities.

### 6.3.3 Populating PRONOIA2

For this case study, the third level of PRONOIA 2 has been populated with previous information. The individuals are shown in Figure 79 and are linked together with properties. Individuals are instances of classes and are linked to classes via properties [Kim et al., 2008]. This figure is composed of nine states represented in the middle. Each state is composed of spatial, temporal and/or spatiotemporal regions and primitives. The different states allow visualizing what happens in the context of AOD and show the product evolution over space and time.

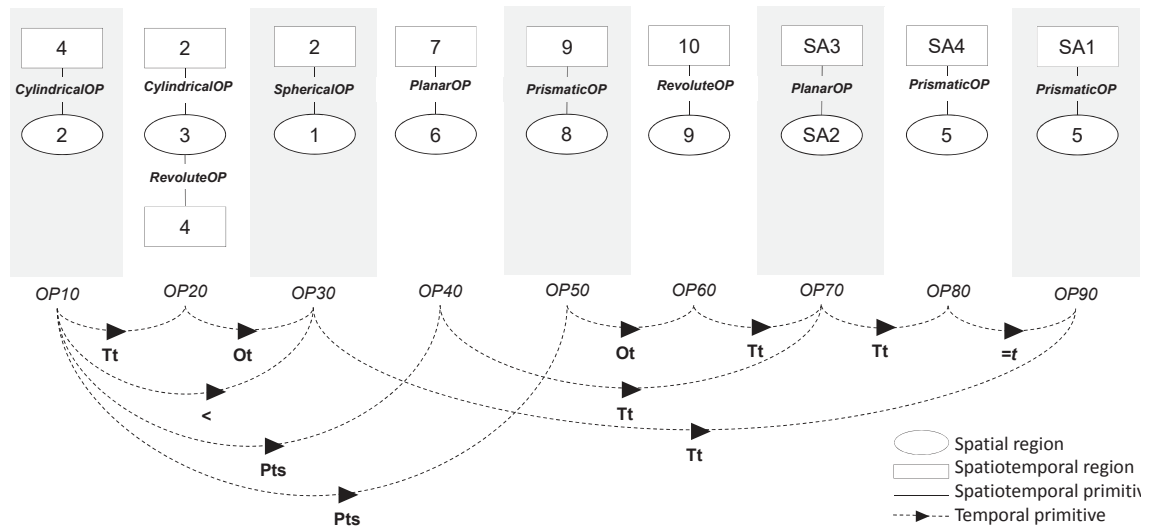


Figure 77: Description of the whole assembly process in a spatiotemporal graph for the second case study

### 6.3.4 Reasoning - consistency checking

After populating the ontology, designers can check that the instance-ontology is composed of all concepts needed in the description of product-process evolution through AOD as well as their consistency. Here no inconsistency has been found by the reasoner (cf. Figure 80).

The ontology in its current shape proposes some available queries to designers so as to look for information requests in a collaborative assembly design environment. In that case the ontology allows a better understanding of product architects' intents. In addition the reasoner can query the knowledge base of the ontology and give designers suggestions on how parts will be assembled over time (how are regions in contact?). So it could also provide some design recommendation to support designers. The assembly design browser searches for relevant assembly information by using a keyword (in DL Query in Protégé). Queries 4 and 5 have been written in SPARQL Query that allows to do more complex queries. Some examples of queries are:

- Query 1** Show all swept volumes in the case study;
- Query 2** Show all physical parts in the case study;
- Query 3** Show all *RevoluteOP* in the case study;
- Query 4** Show all swept volumes and their related spatiotemporal primitives;
- Query 5** Show all physical parts and their related spatiotemporal primitives.

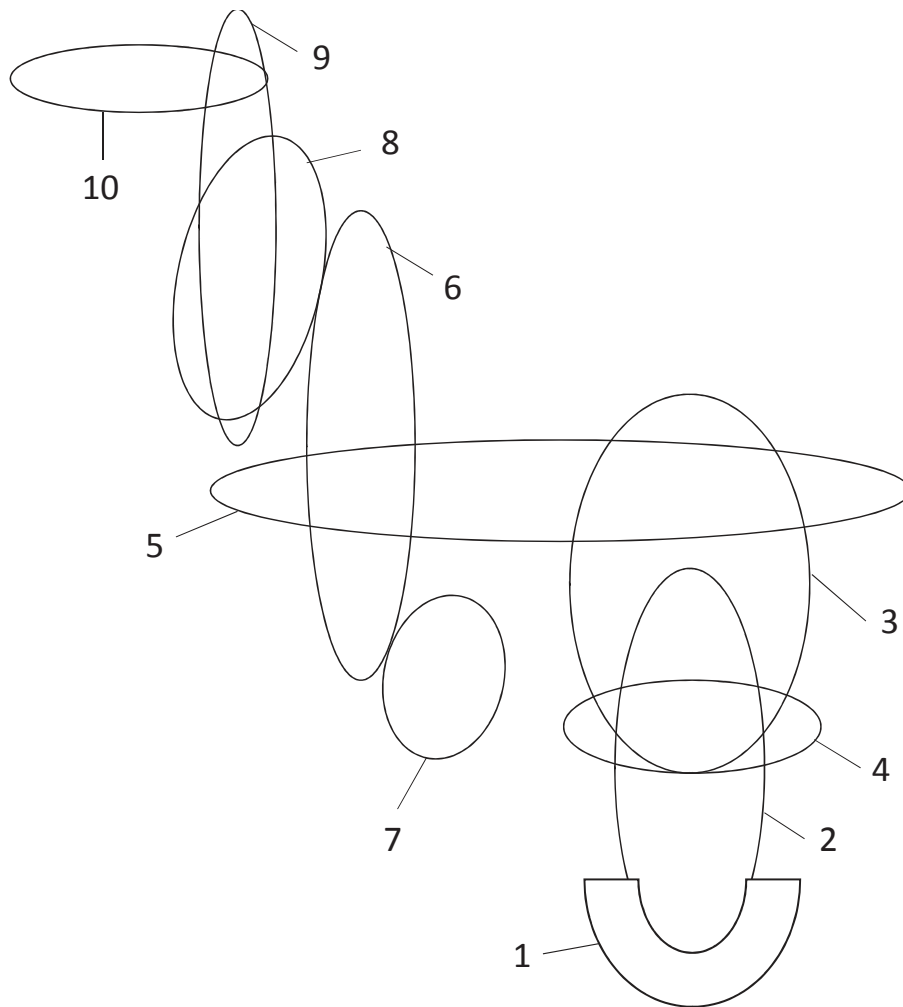


Figure 78: Spatial representation of the whole assembly process for the second case study

The first query and the related answers can be seen in Figure 81 and the ones to the fourth query in Figure 82. By combining the answers of query 4 and query 5, designers can know and understand how parts are related together.

After checking information consistency related to product architect's and the assembly planner's intents [Gruhier et al., 2015a], the designer has a suitable design context based on perdurantist vision to define consistent mechanical assembly.

### 6.3.5 Case study solving following the approach on design evolution

**Create a part** The designer creates the parts of the assembly using a CAD application.

**Define a change** The designer's experience aids him to shortly notice if the product

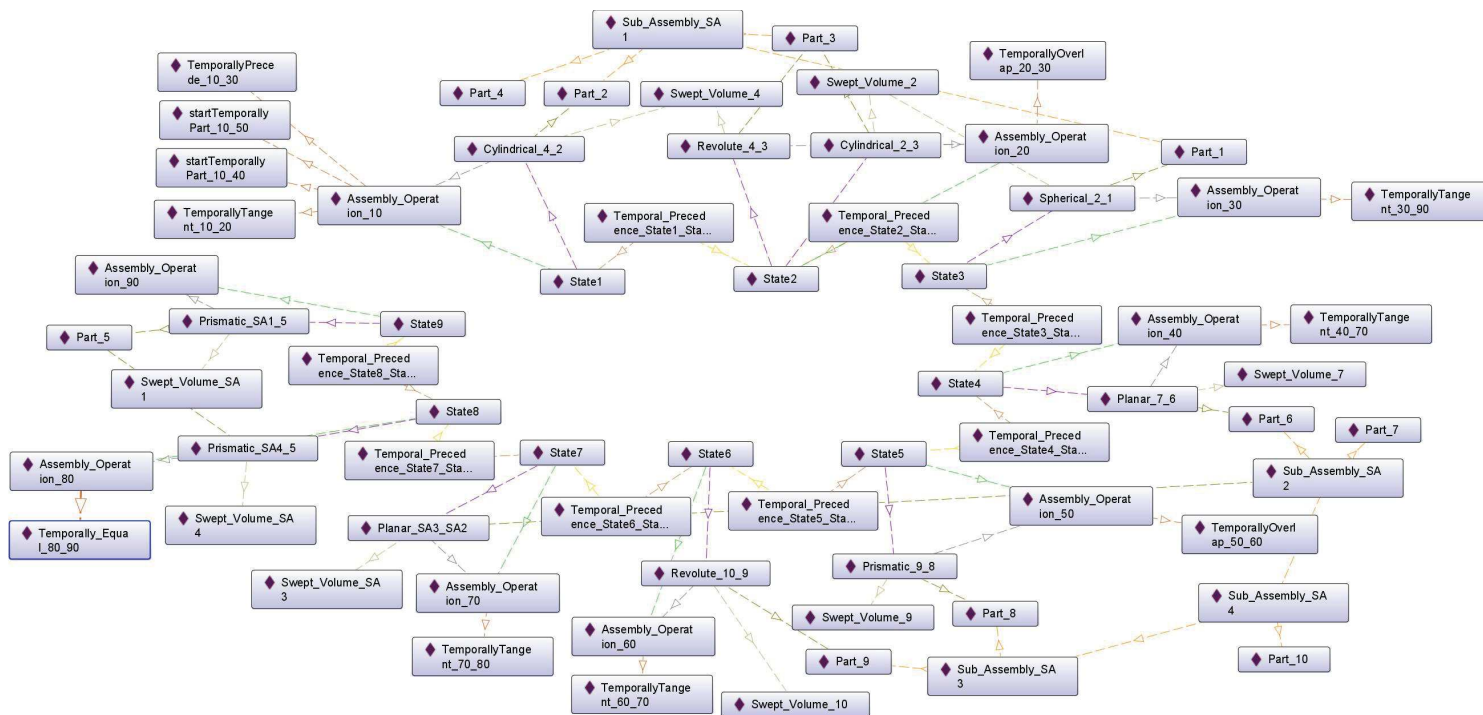


Figure 79: Instances of the ontology with the “kinematic pairs positioning” primitives for the second case study



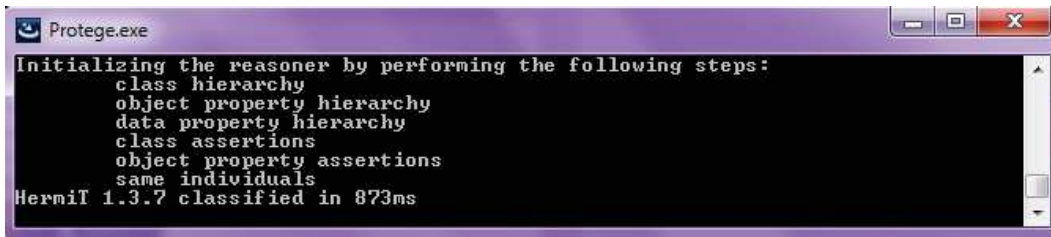


Figure 80: Result of consistency checking of the second case study within Protégé

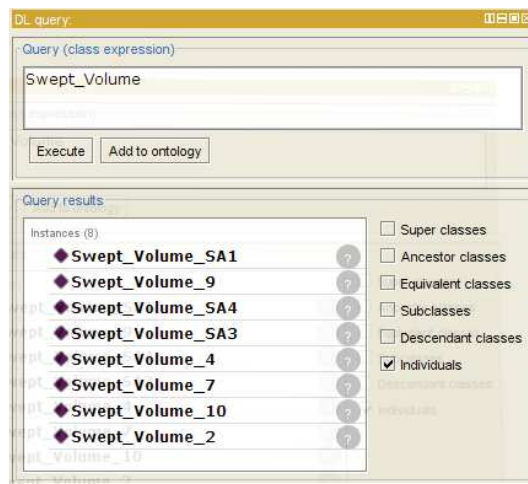


Figure 81: Query “Show Swept Volume” and answers for the second case study

is going to resist deformation or can be simplified to earn time during the manufacturing process. As such, designers may want to change a relationship between parts or define a new one from the “design changes” table previously defined. For instance, here the designer chooses to increase the drone leg dimension to have a better support on the ground. Besides, a change of form of the drone leg in contact with the ground is required by the designer. Then, the ontology is populated (cf. Figure 83 and a consistency checking can be done.

**Undertake the change** After defining what needs to be changed, the designer undertakes the modification in a CAD application.

**Generate the spatiotemporal graph** Then, a spatiotemporal graph (cf. Figure 84) is generated, which sums up all changes undertaken by the designer during the design process.

**Visualize the product** This mechanical assembly, composed of ten parts as listed in Table 42, can then be visualized within a CAD application (e.g. CATIA v5) as illustrated in Table 41 with the different assembly and interface skeletons and the different parts numbers. The case study has only described a part of the whole assembly. Here, the parts 8, 9, 10 and the parts 1, 2, 3, 4 can be copied, as the

SPARQL query:

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX mechanics: <http://www.utbm.fr/mechanics/Ontology.owl#>
SELECT DISTINCT ?Swept_Volume ?Spatiotemporal_Primitive ?type
WHERE
{
    ?Swept_Volume rdf:type mechanics:Swept_Volume .
    ?Spatiotemporal_Primitive mechanics:Spatiotemporal_Primitive_is_Between_Swept_Volume ?Swept_Volume .
    ?Spatiotemporal_Primitive rdf:type ?type
}

```

Swept_Volume	Spatiotemporal_Primitive	type
Swept_Volume_2	Cylindrical_2_3	NamedIndividual
Swept_Volume_2	Cylindrical_2_3	CylindricalOP
Swept_Volume_SA4	Prismatic_SA4_5	NamedIndividual
Swept_Volume_SA4	Prismatic_SA4_5	PrismaticOP
Swept_Volume_10	Revolute_10_9	RevoluteOP
Swept_Volume_10	Revolute_10_9	NamedIndividual
Swept_Volume_SA1	Prismatic_SA1_5	NamedIndividual
Swept_Volume_SA1	Prismatic_SA1_5	PrismaticOP
Swept_Volume_2	Spherical_2_1	NamedIndividual
Swept_Volume_2	Spherical_2_1	SphericalOP
Swept_Volume_SA3	Planar_SA3_SA2	NamedIndividual
Swept_Volume_SA3	Planar_SA3_SA2	PlanarOP
Swept_Volume_4	Revolute_4_3	NamedIndividual

Execute

Figure 82: Query “Show all swept volumes in the second case study and their related spatiotemporal primitives” and answers



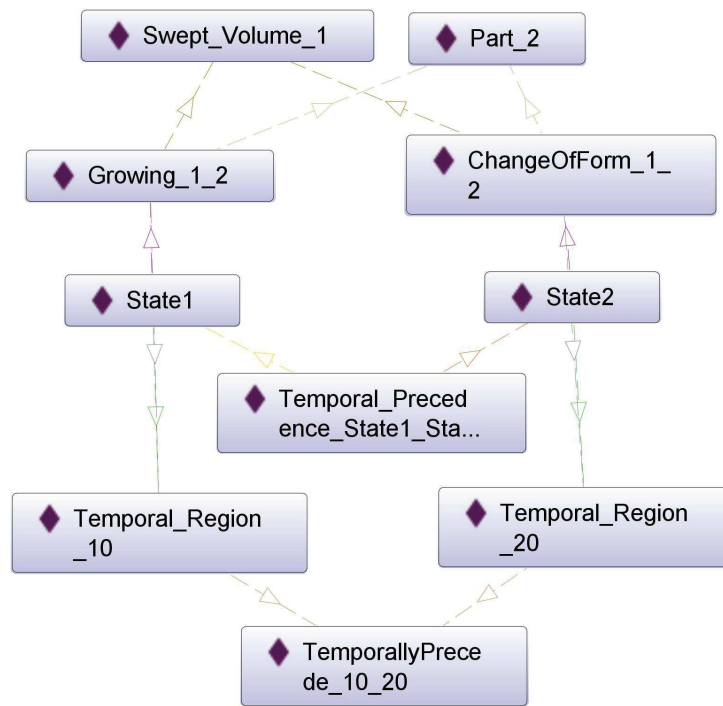


Figure 83: Instances of the ontology with the “design changes” primitives for the second case study

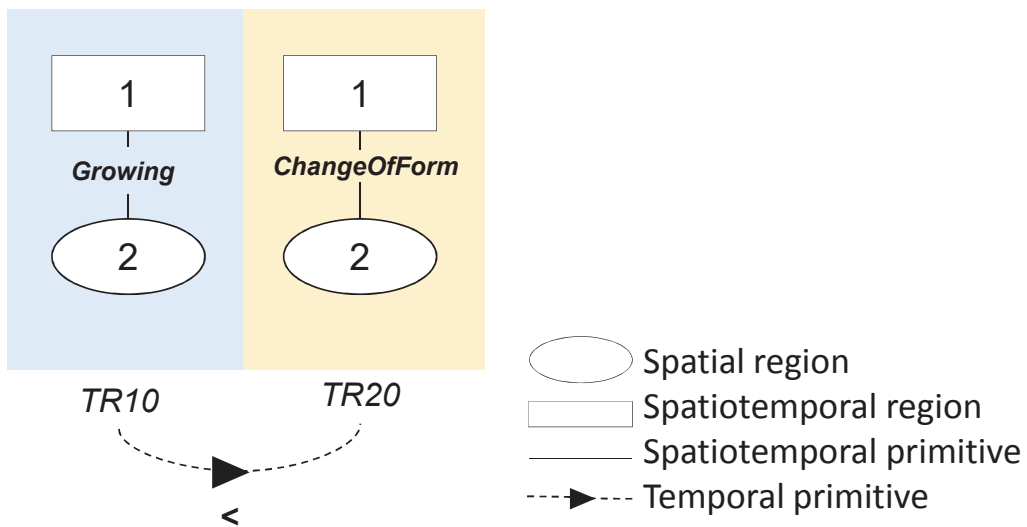


Figure 84: Description of the changes undertaken during the design process in a spatiotemporal graph for the second case study

product has a symmetry. The designer can also add a shell covering the whole system so as to have a fashionable and ergonomic product.

Table 42: Parts list of the second case study

No.	Part name	No.	Part name
1	Drone leg	6	Rack and pinion
2	Piston rod	7	Rack and pinion motor
3	Pneumatic cylinder	8	Cog
4	Spring	9	Propeller support
5	Frame	10	Propeller

The UML sequence diagram (cf. Figure 85) illustrates and sums up the information flow through tasks in Pegasus hub. Consistency is checked through PRONOIA2 ontology each time an arrow goes in the direction of the “Mediator” arrow.

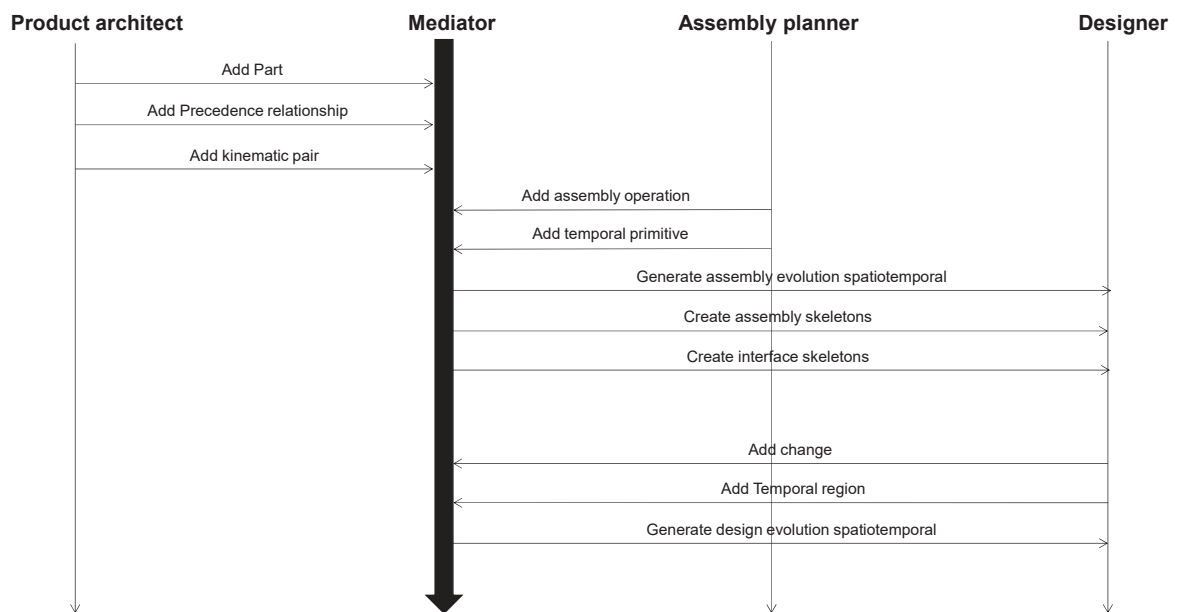


Figure 85: UML diagram for the second case study showing activities and the related stakeholders during the design process

## **6.4 Summary**

In this chapter, our contribution has been applied to two mechanical case studies (i.e. a simplified case study and a medical drone). The last one has been developed following the proposed approach, including the description with the JANUS theory and the reasoning with the PRONOIA2 ontology. Hence, a complete description of the product evolution in relation with its assembly operations, as well as design changes, has been presented with the JANUS theory. Besides, PRONOIA2 ontology has been populated at each step of the approach enabling a formalization of the information and a simultaneously reasoning. Therefore, a consistent product (i.e. consistency of the information given by product architect and assembly planner has been positively checked) has been designed by the designer.

Table 39: Mereotopological description of the second mechanical case study (1)  
**Generate assembly and interface skeletons and the product structure**

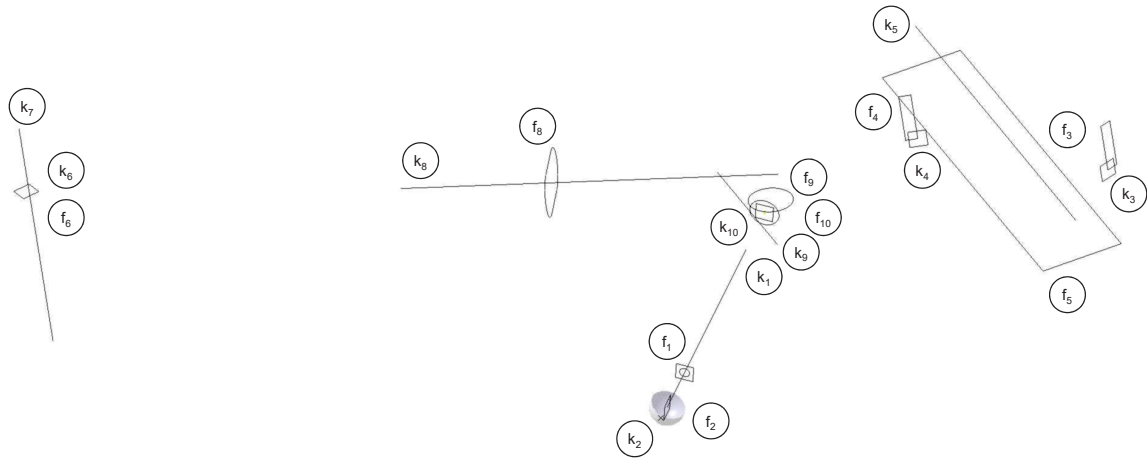
OP No.	Spatiotemporal primitive	Mereotopological description	Skeleton
10	4 <b>CylindricalOP</b> 2	$((2\mathbf{D}4) \wedge (2\mathbf{O}k_1) \wedge (4\mathbf{D}k_1))_{TR1} \wedge (4\mathbf{Move}2)_{TR2} \wedge$ $((2\mathbf{D}4) \wedge (2\mathbf{O}k_1) \wedge (4\mathbf{O}k_1))_{TR3} \wedge (4\mathbf{Move}2)_{TR4} \wedge$ $((2\mathbf{X}4) \wedge (2\mathbf{O}k_1) \wedge (4\mathbf{O}k_1))_{TR5} \wedge (4\mathbf{Move}2)_{TR6} \wedge$ $((2\mathbf{X}4) \wedge (2\mathbf{O}k_1) \wedge (4\mathbf{O}k_1) \wedge (4\mathbf{T}f_1))_{TR7}$	$k_1$ line, $f_1$ surface
20	2 <b>PrismaticOP</b> 3 and 4 <b>RevoluteOP</b> 3	$((3\mathbf{D}2) \wedge (3\mathbf{D}4) \wedge (3\mathbf{O}k_1) \wedge (2\mathbf{D}k_1))_{TR1} \wedge$ $((4\mathbf{Move}3) \wedge (2\mathbf{Move}3))_{TR2} \wedge ((3\mathbf{D}2) \wedge$ $(3\mathbf{O}k_1) \wedge (2\mathbf{O}k_1) \wedge (4\mathbf{O}k_1))_{TR3} \wedge ((4\mathbf{Move}3) \wedge$ $(2\mathbf{Move}3))_{TR4} \wedge ((3\mathbf{X}2) \wedge (3\mathbf{X}4) \wedge (3\mathbf{O}k_1) \wedge (2\mathbf{O}k_1) \wedge$ $(4\mathbf{O}k_1))_{TR5} \wedge ((4\mathbf{Move}3) \wedge (2\mathbf{Move}3))_{TR6} \wedge$ $((3\mathbf{T}4) \wedge (3\mathbf{X}4) \wedge (3\mathbf{O}k_1) \wedge (3\mathbf{T}k_{10}) \wedge (3\mathbf{T}f_{10}) \wedge$ $(2\mathbf{O}k_1) \wedge (4\mathbf{O}k_1) \wedge (4\mathbf{T}k_{10}) \wedge (4\mathbf{T}f_{10}) \wedge (2\mathbf{T}f_{10}))_{TR7}$	$k_1$ line, $k_{10}$ plane $f_{10}$ surface
30	2 <b>SphericalOP</b> 1	$(1\mathbf{D}2)_{TR1} \wedge (2\mathbf{Move}1)_{TR2} \wedge ((1\mathbf{St}2) \wedge (1\mathbf{IP}2) \wedge$ $(1\mathbf{T}f_2) \wedge (2\mathbf{T}f_2) \wedge (1\mathbf{O}k_2) \wedge (2\mathbf{O}k_2))_{TR3}$	$k_2$ point, $f_2$ surface
40	7 <b>PlanarOP</b> 6 and 5 <b>PlanarOP</b> 4	$((6\mathbf{D}7) \wedge (6\mathbf{T}k_3))_{TR1} \wedge (7\mathbf{Move}6)_{TR2} \wedge ((6\mathbf{T}7) \wedge$ $(6\mathbf{T}f_3) \wedge (7\mathbf{T}k_3) \wedge (f_3\mathbf{T}k_3))_{TR3}$	$k_3$ plane, $f_3$ surface
50	9 <b>PrismaticOP</b> 8	$((8\mathbf{D}9) \wedge (8\mathbf{O}k_8) \wedge (9\mathbf{D}k_8))_{TR1} \wedge (9\mathbf{Move}8)_{TR2} \wedge$ $((8\mathbf{D}9) \wedge (8\mathbf{O}k_8) \wedge (9\mathbf{O}k_8))_{TR3} \wedge (9\mathbf{Move}8)_{TR4} \wedge$ $((8\mathbf{X}9) \wedge (8\mathbf{O}k_8) \wedge (9\mathbf{O}k_8))_{TR5} \wedge (9\mathbf{Move}8)_{TR6} \wedge$ $((8\mathbf{X}9) \wedge (8\mathbf{O}k_8) \wedge (9\mathbf{O}k_8) \wedge (9\mathbf{T}f_8))_{TR7}$	$k_8$ line, $f_8$ surface

Table 40: Mereotopological description of the second mechanical case study (2)

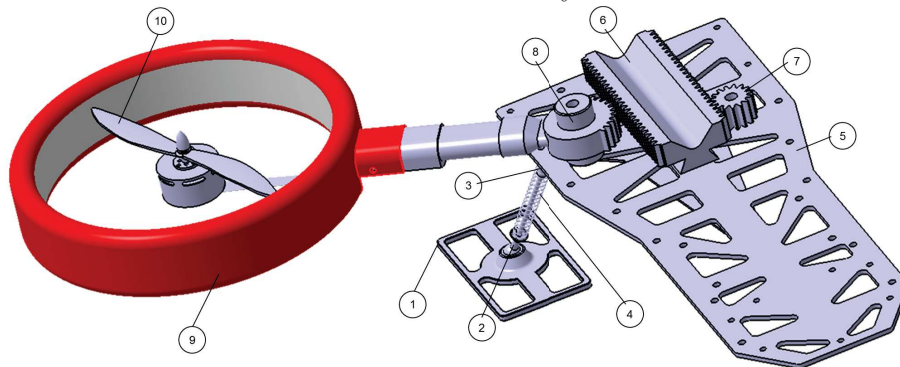
OP No.	Spatiotemporal primitive	Mereotopological description	Skeleton
60	10 <b>RevoluteOP</b> 9	$((9\mathbf{D}10) \wedge (9\mathbf{O}k_6) \wedge (10\mathbf{D}k_6))_{TR1} \wedge (10\mathbf{Move}9)_{TR2} \wedge$ $((9\mathbf{D}10) \wedge (9\mathbf{O}k_6) \wedge (10\mathbf{O}k_6))_{TR3} \wedge (10\mathbf{Move}9)_{TR4} \wedge$ $((9\mathbf{X}10) \wedge (9\mathbf{O}k_6) \wedge (10\mathbf{O}k_6))_{TR5} \wedge (10\mathbf{Move}9)_{TR6} \wedge$ $((9\mathbf{T}10) \wedge (9\mathbf{X}10) \wedge (9\mathbf{O}k_6) \wedge (9\mathbf{T}k_7) \wedge (9\mathbf{T}f_6) \wedge$ $(10\mathbf{O}k_6) \wedge (10\mathbf{T}k_7) \wedge (10\mathbf{T}f_6))_{TR7}$	$k_6$ line, $k_7$ plane, $f_6$ surface
70	$SA3$ <b>PlanarOP</b> $SA2$	$((SA2\mathbf{D}SA3) \wedge (SA2\mathbf{T}k_4))_{TR1} \wedge$ $(SA3\mathbf{Move}SA2)_{TR2} \wedge ((SA2\mathbf{T}SA3) \wedge (SA2\mathbf{T}f_4) \wedge$ $(SA3\mathbf{T}k_4) \wedge (f_4\mathbf{T}k_4))_{TR3}$	$k_4$ plane, $f_4$ surface
80	$SA4$ <b>PrismaticOP</b> 5	$((5\mathbf{D}SA4) \wedge (5\mathbf{O}k_5) \wedge (SA4\mathbf{D}k_5))_{TR1} \wedge$ $(SA4\mathbf{Move}5)_{TR2} \wedge ((5\mathbf{D}SA4) \wedge (5\mathbf{O}k_5) \wedge$ $(SA4\mathbf{O}k_5))_{TR3} \wedge (SA4\mathbf{Move}5)_{TR4} \wedge ((5\mathbf{X}SA4) \wedge$ $(5\mathbf{O}k_5) \wedge (SA4\mathbf{O}k_5))_{TR5} \wedge (SA4\mathbf{Move}5)_{TR6} \wedge$ $((5\mathbf{X}SA4) \wedge (5\mathbf{O}k_5) \wedge (SA4\mathbf{O}k_5) \wedge (SA4\mathbf{T}f_5))_{TR7}$	$k_5$ line, $f_5$ surface
90	$SA1$ <b>PrismaticOP</b> 5	$((5\mathbf{D}SA1) \wedge (5\mathbf{O}k_9) \wedge (SA1\mathbf{D}k_9))_{TR1} \wedge$ $(SA1\mathbf{Move}5)_{TR2} \wedge ((5\mathbf{D}SA1) \wedge (5\mathbf{O}k_9) \wedge$ $(SA1\mathbf{O}k_9))_{TR3} \wedge (SA1\mathbf{Move}5)_{TR4} \wedge ((5\mathbf{X}SA1) \wedge$ $(5\mathbf{O}k_9) \wedge (SA1\mathbf{O}k_9))_{TR5} \wedge (SA1\mathbf{Move}5)_{TR6} \wedge$ $((5\mathbf{X}SA1) \wedge (5\mathbf{O}k_9) \wedge (SA1\mathbf{O}k_9) \wedge (SA1\mathbf{T}f_9))_{TR7}$	$k_9$ line, $f_9$ surface

Table 41: Second case study representation

## Assembly and interface skeletons



## Studied assembly



## Whole assembly





# Chapter 7

## Conclusions and future work

### Contents

---

<b>7.1</b>	<b>Overall conclusion . . . . .</b>	<b>148</b>
<b>7.2</b>	<b>Future work . . . . .</b>	<b>149</b>
7.2.1	Short-term future work: CAD support improving designer's understanding . . . . .	149
7.2.2	Mid-term future work: Transformable products description . . . . .	150
7.2.3	Long-term future work: Consideration of other lifecycle stages and domains . . . . .	151
<b>1</b>	<b>Research work publications . . . . .</b>	<b>153</b>
1.1	Peer-reviewed international journals . . . . .	153
1.2	Peer-reviewed international conferences . . . . .	153
1.3	Peer-reviewed national symposium . . . . .	154
<b>2</b>	<b>Other references . . . . .</b>	<b>154</b>

---



The purpose of this chapter is to summarize our scientific contribution and introduce some future work propositions. The first sub-section is the general conclusion, which reminds the main contribution throughout the different thesis chapters. The second sub-section defines how the research work could be enhanced and the domains in which it could be extended.

## 7.1 Overall conclusion

The research work brings a novel and innovative contribution in the domain of mereotopology, ontology and product-process information management approach. The contribution focuses on design and assembly phases so as to support concurrent and proactive engineering. That is the reason why the objective is to propose a product-process information management approach to control assembly and design evolution and provide a complete understanding to product architects and designers.

The first chapter has presented the industrial and scientific contexts in order to define the boundary of this research work, as well as to list the requirements not yet fulfilled in the design process at the beginning of the PhD.

In the second chapter, a strong state of the art has been presented related to mereotopological theories, spatiotemporal ontologies and approaches managing product evolution or changes. Built on this, each point of our research works have positioned according to already published works. Here, a lack of approach to improve product architects' and designers' understanding during the early product development process have been highlighted. Moreover, products are nowadays only considered from an endurantist point of view and not from a perdurantist one, which considers product evolution.

The third chapter has introduced the first part of our contribution: the JANUS theory. This novel mereotopological theory covers three dimensions (i.e. spatial, temporal and spatiotemporal). Indeed, new spatiotemporal objects and primitives have been introduced in order to describe product-process knowledge and information over space and time. Here a philosophy of perdurantism is considered. By considering the spatiotemporal dimension and the associated mereotopology-based theory, information has been added to technical objects so as to express their history in a consistent and understandable manner. This description aids to understand phenomena during AOD process. For instance, product evolutions over time are captured at each change, enabling a better understanding of the product. The first part of our contribution is considered here as the basis for future studies.

In the fourth chapter, we have presented the second part of our contribution: the PRONOIA2 ontology. The ontology formalizes the JANUS theory with OWL-DL and SWRL languages and makes it machine-interpretable for PLM systems. As such, by considering these novel rules, a reasoning layer has been implemented so as to reason on spatiotemporal associations. Such efforts enable the introduction of novel procedures for consistency checking of product-process information and knowledge in PLM, and create

---

a robust epistemological and methodological platform.

The fifth chapter has presented the third part of our contribution: a novel relationships management approach to describe product evolution, called MERCURY. An improved hub (i.e. Pegasus) has been introduced so as to orchestrate spatial and temporal information in the upstream design process. Indeed, the approach enables a mechanism of reconciliation between PDM and MPM systems. Designer is supported in his tasks by this relevant information in the early design stages. Pegasus advantages are to capture new information, keep a traceability of the assembly and design evolution and orchestrate information flow between design and manufacturing. Moreover, product evolution over its lifecycle in a collaborative context is supported, as the product architect, the assembly planner and the designer use common semantic. Therefore, concurrent and proactive engineering are ensured as the product evolution is controlled along its lifecycle, as well as the information flow between stakeholders.

In the sixth chapter, we have applied the approach to two mechanical case studies in order to show the relevance of the proposed theory, ontology and approach. The first academic case presents the application of the mereotopology theory and introduces new entities such as the swept volumes. The second case is more focused on the application of the proposed approach.

To summarize, our contribution – on product-process spatiotemporal qualitative description – is composed of a theory based on mereotopology, an ontology model and a management approach. The actual stake is to get a long term dynamic vision of the space over time in order to facilitate the understanding of assembly and design changes. Our contribution enables providing product architect and designer awareness and understanding about each other intents so as to efficiently design a consistent product. Therefore, this thesis introduces a novel paradigm in integrated design based on perdurantism.

## 7.2 Future work

Multiple further efforts could be developed in different domains. Here we focus and stay in the mechanical engineering domain. Three future works are presented: a short-, middle- and long-term works.

### 7.2.1 Short-term future work: CAD support improving designer's understanding

In this section, short-term future work is presented, which directly follows previous works. After describing the product with mereotopology, the objective here will be to use this information in a future CAD tools. Indeed, current CAD tools can only design spatial product [Bauzer Medeiros, 2002] and represent the final physical model [Fischer and McKinney Liston, 2001]. The representation of static products is currently the major drawback of 3D definition. It does not represent the product as it looks like in the reality. The gap

between 3D model and real object is going to be narrowed and model will behave as in reality.

As such, geometric evolution tools will evolve: 3D design will be transformed to 4D design in order to get closer to the reality, which is composed of dynamic products. Indeed, 4D design can show the product evolution along its assembly phases. The development of a CAD support for designers will enable controlling the product geometry definition and its temporal evolution based on assembly planning. The ability of future CAD to include manufacturing phases enables linking spatial and temporal dimensions [Friedlander, 2009]. Therefore this new spatiotemporal dimension improves the understanding of a 3D CAD model by adding an information layer.

Future CAD will be based on relationships between parts and not on parts. Product evolution will be considered during design process and will be represented with spatiotemporal regions. Moreover, 4D design will enable improving designer's understanding, as he will explicitly understand design and assembly intents.

This future work will be based on a skeleton-based approach in order to define an assembly context for product development process. This approach will for instance enable the dynamic representation of assembly and interface skeletons according to design intents. As such, the designer will have a better understanding of how skeletons, on which the design is based, evolve, move and are deformed during the assembly process. Indeed, product will be designed at different states (defined according to the mereotopological description of the spatiotemporal primitives). Moreover, parts will be simultaneously checked if they have been designed to respect the kinematic pairs defined by the product architect.

### **7.2.2 Mid-term future work: Transformable products description**

Current CAD tools support the design of mechanical parts. Indeed, designers can just create static spatial objects [Kiritsis, 2011]. As such, current CAD tools are not suited to design transformable products, which have the ability to transform [Singh et al., 2009]. Current transformable products are created on two different static CAD models. CAD tools do not give any information on the different states of the transformation and how the product is changing from one configuration to another [Singh et al., 2007]. CAD tools limits could be overcome by adding a dynamic CAD application taking into consideration the product evolution at the early design stages [Kiritsis, 2011].

Besides, product types are evolving over time. The first type of product used in industry was the basic mechanical product. Then, mechatronic products have arrived on the market. Different domains, such as mechanical and electrical, are concerned with mechatronic products. Connected products have recently been introduced in our everyday life. As such, products are linked to computers. Moreover, transformable products

are currently developed in industry and have the ability to change their configurations over time during the use phase. This type of product must adapt to an environment. The next step is to deal with intelligent product composed of smart and programmable matter.

Therefore this middle-term future work focuses on a novel approach to support designers at the early design stages to design transformable products, the new generation of product. Working with transformable products is the step before being able to work with intelligent products, which can self-adapt to their changing environments. Although intelligent products are well-known in science-fiction, they are beginning to enter our everyday life. So this new generation requires new technological challenges to be designed [Kiritsis, 2011]. As such product architect, assembly planner and designer need to change how they work to design a product, in order to maximize the benefits of 4D technologies. The product evolution during the assembly and the use phases will be described based on spatiotemporal relationships between parts.

### 7.2.3 Long-term future work: Consideration of other lifecycle stages and domains

This long-term future work is divided in two parts: the applications in the domain of engineering design and the ones in other domains.

The different product lifecycle stages, where our research work could be adapted, are presented in a sequential manner. Each stage will now be explained into details.

**Requirement** Mereotopology could be used at the early stages of the product lifecycle, especially in the requirement process. In that case, needs could be translated not only into functions, but also in spatiotemporal mereotopological primitives. For instance, if the need is to “have a wheel which could turn in relation with the car”, could be translated by a *RevoluteOP*.

**Design** During the design phase, mereotopology could be useful to facilitate the designer’s work. Indeed, when the product architect chooses the kinematic pair between two parts, the designer could have access to a virtual library showing him different solutions in accordance to the specified spatiotemporal primitives. As such, the designer’s work could be carried out faster.

**Use** Mereotopology could also be used to describe product deformation and so on during the use phase. As such, relationships between objects could be studied when a force or another object is applied on it. In that case, behavioral laws should be known to predict the importance of the deformation on the product.

**Maintenance** Mereotopology could be used in order to design product, which could be easily and fast dismantled to change a broken part or a consumable

**Disassembly** The same approach, as the one described in this thesis, could be extended to the disassembly process. In that case, new constraints will be introduced and the

mereotopological primitives should be reversed. In fact, the parts will be dismantled and not inserted on other parts. A method for finding disassembly sequence should be defined, such as the one developed by Chekh Wais [Chekh Wais et al., 2014].

In parallel of this lifecycle, mereotopology could be applied to materials. Each material has specific features such as Young modulus, Poisson's ratio and so on. Mereotopology could aid to describe the position of a material related to another. For instance, the user could query for a specific feature and the system would give all material families closed to the requirement.

The proposed research work could also have an application in other domains, in which objects (e.g. *Spime* in the domain of IoT) evolve over space and time. The purpose could be to have a traceability of information or aid for predictions, such as:

- GIS (e.g. description of landscape evolution over time [Del Mondo, 2011]);
- Urban design (e.g. description of the city evolution over time like the creation of buildings [Lu et al., 2009]);
- Medical science (e.g. description of diseases and comparison with the current state of patient to detect potential illness [Strange et al., 2014]);
- Heritage site and historical building (e.g. description of the heritage in order to keep a traceability, save the heritage [Laroche, 2007] and predict future technics which could be developed, as well as represent the evolution [Stefani et al., 2009]);
- And so on.

# References

## 1 Research work publications

### 1.1 Peer-reviewed international journals

[Gruhier et al., 2015a] **Gruhier E.**, Demoly F., Dutartre O., Abouddi S., Gomes S., *A formal ontology-based spatiotemporal mereotopology for integrated assembly design and assembly sequence planning*, Advanced Engineering Informatics, 29:495-512, 2015, doi 10.1016/j.aei.2015.04.004, 2015

[Gruhier et al., 2015b] **Gruhier E.**, Demoly F., Kim K.-Y., Abouddi S., Gomes S., *A theoretical framework for product relationships description over space and time in integrated design*, Journal of Engineering Design, Under review, 2015

### 1.2 Peer-reviewed international conferences

[Gruhier, 2014a] **Gruhier E.**, *Towards a qualitative description of spatiotemporal mereotopology-based product relationships*, Product and Asset Lifecycle Management, PALM2014 workshop, 26-29 Janvier, 2014, Les Diablerets, Suisse

[Gruhier et al., 2014b] **Gruhier E.**, Demoly F., Kim K.-Y. Gomes S., *Mereotopology and product design: State of the art and research challenges*, Tools and Methods of Competitive Engineering, TMCE2014, 19-23 Mai, 2014, Budapest, Hongrie, ISBN 978-94-6186-177-1

[Gruhier et al., 2014c] **Gruhier E.**, Demoly F., Abouddi S., Gomes S., *A spatiotemporal mereotopology-based theory for qualitative description in assembly design and sequence planning*, Design Computing and Cognition, DCC2014, 23-25 Juin, 2014, Londres, Royaume-Uni

[Gruhier et al., 2014d] **Gruhier E.**, Demoly F., Dutartre O., Abouddi S., Gomes S., *Towards a spatiotemporal ontology-based on mereotopological theory in assembly-oriented design*, Advances in Production Management Systems, APMS2014, 20-24 Septembre, 2014, Ajaccio, France

### 1.3 Peer-reviewed national symposium

[Gruhier et al., 2014e] **Gruhier E.**, Demoly F., Abboudi S., Gomes S., *Description qualitative basée sur une théorie méreotopologique spatio-temporelle en conception orientée assemblage*, Ingédodc2014, 10 Avril, 2014, Sévenans, France

[Gruhier et al., 2015c] **Gruhier E.**, Demoly F., Abboudi S., Gomes S., *Description qualitative du couple produit-process basée sur la méreotopologie spatio-temporelle*, AIP Priméca2015, 31 Mars - 2 Avril, 2015, La Plagne, France

## 2 Other references

[Aiello et al., 2007] Aiello, M., Pratt-Hartmann, I., and Van Benthem, J. (2007). *Handbook of spatial logics*. Springer.

[Al-Debei et al., 2012] Al-Debei, M. M., Al Asswad, M. M., De Cesar, S., and Lycett, M. (2012). Conceptual modelling and the quality of ontologies : endurantism vs perdurantism. *International Journal of Database Management Systems*, 4(3):1–19.

[Allen, 1983] Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

[Antoniou and Van Harmelen, 2008] Antoniou, G. and Van Harmelen, F. (2008). *A semantic Web primer*. MIT Press, Cambridge, Mass., second edition.

[Arenas Solutions, 2015] Arenas Solutions (2015). Three tips for creating a manufacturing change process that works.

[Artale et al., 1996] Artale, A., Franconi, E., Guarino, N., and Pazzi, L. (1996). Part-whole relations in object-centered systems: An overview. *Data and Knowledge Engineering*, 20(3):347–383.

[Asher and Vieu, 1995] Asher, N. and Vieu, L. (1995). Toward a geometry of common sense: A semantics and a complete axiomatization of mereotopology. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 846–852.

[Baader et al., 2007] Baader, F., Calvanese, D., McGuinness, D., Nardi, D., and Patel-Schneider, P. (2007). *The description logic handbook - second edition*. Cambridge University Press.

[Baldwin et al., 1998] Baldwin, A., Austin, S., Hassan, T., and Thorpe, A. (1998). Planning building design by simulating information flow. *Automation in Construction*, 8:149–163.

[Batres et al., 2007] Batres, R., West, M., Leal, D., Price, D., and Naka, Y. (2007). An upper ontology based on ISO 15926. *Computers and Chemical Engineering*, 31:519–534.



- [Batsakis and Petrakis, 2011] Batsakis, S. and Petrakis, E. (2011). SOWL: a framework for handling spatio-temporal information in OWL 2.0. In *Rule-Based Reasoning, Programming, and Applications*, pages 242–249. Springer.
- [Bauzer Medeiros, 2002] Bauzer Medeiros, C. (2002). Spatiotemporal information systems.
- [Bennett et al., 2000] Bennett, B., Cohn, A. G., Torrini, P., and Hazarika, S. (2000). Describing rigid body motions in a qualitative theory of spatial regions. In *Proceedings of the National Conference on Artificial Intelligence*, pages 503–509.
- [Bergson, 1923] Bergson, H. (1923). *Creative evolution*. H. Holt and Company.
- [Bittner, 2001] Bittner, T. (2001). Rough sets in spatio-temporal data mining. In *Temporal, spatial, and spatio-temporal data mining*, pages 89–104. Springer.
- [Bittner et al., 2006] Bittner, T., Donnelly, M., and Smith, B. (2006). A spatio-temporal ontology for geographic information integration. *International Journal of Geographical Information Science*, 23(6):765–798.
- [Boothroyd et al., 2002] Boothroyd, G., Dewhurst, P., and Knight, W. (2002). *Product design for manufacture and assembly*. Taylor and Francis, second edition.
- [Borgo et al., 1996] Borgo, S., Guarino, N., and Masolo, C. (1996). A pointless theory of space based on strong connection and congruence. In *Principles of Knowledge Representation and Reasoning-International Conference*, pages 220–229.
- [Borgo and Masolo, 2008] Borgo, S. and Masolo, C. (2008). Full mereogeometries. *Journal of Philosophical Logic*, 1:1–46.
- [Bourjault, 1984] Bourjault, A. (1984). *Contribution à une approche méthodologique de l'assemblage automatisé; élaboration automatique des séquences opératoires*. PhD thesis, Besançon.
- [Bracewell et al., 2009] Bracewell, R., Wallace, K., Moss, M., and Knott, D. (2009). Capturing design rationale. *Computer-Aided Design*, 41(3):173–186.
- [Brown, 2006] Brown, J. (2006). Managing product relationships Enabling iteration and innovation in design. Technical report, AberdeenGroup.
- [Card et al., 1999] Card, S., MacKinlay, J., and Schneiderman, B. (1999). *Information visualization using vision to think*. Morgan Kaufmann.
- [Casati and Varzi, 1999] Casati, R. and Varzi, A. C. (1999). *Parts and places: The structures of spatial representation*. MIT press.
- [Chandrasegaran et al., 2013] Chandrasegaran, S., Ramani, K., Sriram, R. D., Horváth, I., Bernard, A., Harik, R. F., and Gao, W. (2013). The evolution, challenges, and future of knowledge representation in product design systems. *Computer-Aided Design*, 45(2):204–228.



- [Chang, 2008] Chang, X. (2008). *Ontology Development and Utilization in Product Design*. PhD thesis, Virginia Polytechnic Institute and State University.
- [Chekh Wais et al., 2014] Chekh Wais, H., Mitrouchev, P., and Tollenaere, M. (2014). Disassembly sequencing for end-of-life products. Toulouse, France.
- [Chen and Luo, 2014] Chen, L. and Luo, H. (2014). A BIM-based construction quality management model and its applications. *Automation in Construction*, 46:64–73.
- [Chen et al., 2010] Chen, X., Gao, S., Yang, Y., and Zhang, S. (2010). Multi-level assembly model for top-down design of mechanical products. *Computer-Aided Design*.
- [Christophe et al., 2010] Christophe, F., Bernard, A., and Coatanéa, E. (2010). *RFBS: A model for knowledge representation of conceptual design*. *CIRP Annals - Manufacturing Technology*, 59:155–158
- [Chungoora et al., 2013] Chungoora, N., Young, R., Gunendran, G., Palmer, C., Usman, Z., Anjum, N. A., Cutting-Decelle, A.-F., Harding, J. A., and Case, K. (2013). A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. *Computers in Industry*, 64(4):392–401.
- [CIMdata, 2015] CIMdata (2015). Collaborative manufacturing: Critical for success in today’s complex heavy equipment world.
- [Ciocoiu et al., 2001] Ciocoiu, M., Nau, D. S., and Gruninger, M. (2001). Ontologies for Integrating Engineering Applications. *Journal of Computing and Information Science in Engineering*, 1(1):12.
- [Clarke, 1981] Clarke, B. (1981). A calculus of individuals based on connection. *Notre Dame Journal of Formal Logic*, 22(3):204–218.
- [Clarke, 1985] Clarke, B. L. (1985). Individuals and points. *Notre Dame Journal of formal logic*, 26(1):61–75.
- [Colombo et al., 2007] Colombo, G., Mosca, A., and Sartori, F. (2007). Towards the design of intelligent CAD systems: An ontological approach. *Advanced Engineering Informatics*, 21(2):153–168.
- [David and Rowe, 2015] David, M. and Rowe, F. (2015). What does PLMS (product lifecycle management systems) manage data or documents complementarity and contingency for SMEs. *Computers in Industry*.
- [De Fazio and Whitney, 1987] De Fazio, T. and Whitney, D. E. (1987). Simplified generation of all mechanical assembly sequences. *IEEE Journal of Robotics and Automation*, 3(6):640–658.
- [De Laguna, 1922] De Laguna, T. (1922). Point, line, and surface, as sets of solids. *The Journal of Philosophy*, 19(17):449–461.

- [Del Mondo, 2011] Del Mondo, G. (2011). *Un modèle de graphe spatio-temporel pour représenter l'évolution d'entités géographiques*. PhD thesis, Université de Bretagne occidentale-Brest.
- [Del Mondo et al., 2010] Del Mondo, G., Stell, J. G., Claramunt, C., and Thibaud, R. (2010). A graph model for spatiotemporal evolution. *Journal of universal computer science*, 16(11):1452–1477.
- [Demoly, 2010] Demoly, F. (2010). *Conception intégrée et gestion d'informations techniques: application à l'ingénierie du produit et de sa séquence d'assemblage*. PhD thesis, Université de Technologie de Belfort-Montbéliard.
- [Demoly et al., 2012a] Demoly, F., Deniaud, S., and Gomes, S. (2012a). Towards an harmonious and integrated management approach for lifecycle planning. In *International Conference on Advanced Production Management Systems*, volume 398, pages 208–215, Greece.
- [Demoly et al., 2013a] Demoly, F., Dutartre, O., Yan, X.-T., Eynard, B., Kiritsis, D., and Gomes, S. (2013a). Product relationships management enabler for concurrent engineering and product lifecycle management. *Computers in Industry*, 64(7):833–848.
- [Demoly et al., 2012b] Demoly, F., Matsokis, A., and Kiritsis, D. (2012b). A mereotopological product relationship description approach for assembly oriented design. *Robotics and Computer-Integrated Manufacturing*, 28(6):681–693.
- [Demoly et al., 2012c] Demoly, F., Matsokis, A., Kiritsis, D., and Gomes, S. (2012c). Mereotopological description of product-process information and knowledge for PLM. In *PLM*, volume 388 of *IFIP Advances in Information and Communication Technology*, pages 70–84. Springer.
- [Demoly et al., 2010] 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.
- [Demoly et al., 2013b] Demoly, F., Pels, H. J., and Gomes, S. (2013b). Proactive Engineering and PLM: Current Status and Research Challenges. In *Product Lifecycle Management for Society*, pages 170–181. Springer.
- [Demoly et al., 2011a] Demoly, F., Toussaint, L., Eynard, B., Kiritsis, D., and Gomes, S. (2011a). Geometric skeleton computation enabling concurrent product engineering and assembly sequence planning. *Computer-Aided Design*, 43(12):1654–1673.
- [Demoly et al., 2012d] Demoly, F., Yan, X.-T., Eynard, B., Gomes, S., and Kiritsis, D. (2012d). Integrated product relationships management: a model to enable concurrent product design and assembly sequence planning. *Journal of Engineering Design*, 23(7):544–561.

- [Demoly et al., 2011b] Demoly, F., Yan, X. T., Eynard, B., Rivest, L., and Gomes, S. (2011b). An assembly oriented design framework for product structure engineering and assembly sequence planning. *Robotics and Computer-Integrated Manufacturing*, 27(1):33–46.
- [Descartes, 1996] Descartes, R. (1996). *Meditations on First Philosophy With Selections from the Objections and Replies*. John Cottingham, cambridge university press edition.
- [Dong et al., 2007] Dong, T., Tong, R., Zhang, L., and Dong, J. (2007). A knowledge-based approach to assembly sequence planning. *The International Journal of Advanced Manufacturing Technology*, 32(11-12):1232–1244.
- [Duntsch et al., 2001] Düntsch, I., Wang, H., and McCloskey, S. (2001). A relation-algebraic approach to the region connection calculus. *Theoretical Computer Science*, 255:63–83.
- [Eschenbach, 1999] Eschenbach, C. (1999). A predication calculus for qualitative spatial representations. In *Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science*, pages 749–749, Berlin. Springer.
- [Eschenbach et al., 1994] Eschenbach, C., Habel, C., and Smith, B. (1994). A Mereotopological Definition of Point. *Journal of Philosophical Logic*, 32(4):357–390.
- [Eynard, 2005] Eynard, B. (2005). Gestion du cycle de vie des produits et dynamique des connaissances industrielles en conception intégrée. Habilitation à Diriger les Recherches, Laboratoire des Systèmes Mécaniques et d’Ingénierie Simultanée, Troyes.
- [Fei et al., 2010] Fei, G., Gao, J., Owodunni, O. O., and Tang, X. Q. (2010). A methodology for engineering design change management using modelling and problem solving techniques. In *Proceedings of the 36th International MATADOR Conference*, pages 179–182. Springer.
- [Fensel et al., 2007] Fensel, D., Lausen, H., Polleres, A., de Bruijn, J., Stollberg, M., Roman, D., and Domingue, J. (2007). *Enabling Semantic Web Services. The web service modeling ontology*. Springer verlag edition.
- [Fenves et al., 2008a] Fenves, S., Foufou, S., Bock, C., and Sriram, R. (2008a). CPM2: a core model for product data. *Journal of Computing and Information Science in Engineering*, 8(1):14501.
- [Fenves et al., 2008b] Fenves, S. J., Foufou, S., Bock, C., and Sriram, R. D. (2008b). CPM: A core model for product data. *Journal of computing and information science in engineering*, 5:238–246.
- [Fernandez et al., 1997] Fernandez, M., Gomez-Perez, A., and Juristo, N. (1997). Methontology From ontological art towards ontological engineering. Technical report.
- [Filos, 2012] Filos, E. (2012). The future of PLM. Experiences and Objectives of the European Research Programme.

- [Fiorentini et al., 2008] Fiorentini, X., Rachuri, S., Mahesh, M., Fenves, S. J., and Sriram, R. D. (2008). An evaluation of description logic for the development of product models. Technical report, National Institute of Standards and Technology.
- [Fiorentini et al., 2010] Fiorentini, X., Rachuri, S., Suh, H., Lee, J., and Sriram, R. (2010). An Analysis of Description Logic Augmented with Domain Rules for the Development of Product Models. *Journal of Computing and Information Science in Engineering*, 10(2):021008.
- [Fischer and McKinney Liston, 2001] Fischer, M. and McKinney Liston, K. (2001). 4d CAD Research - Wish list for 4d environments.
- [Fonseca et al., 2000] Fonseca, F., Egenhofer, M., Davis Jr, C., and Borges, K. (2000). Ontologies and knowledge sharing in urban GIS. *Computers, Environment and Urban systems*, (24):251–271.
- [Fortineau et al., 2013] Fortineau, V., Paviot, T., and Lamouri, S. (2013). Improving the interoperability of industrial information systems with description logic-based models - The state of the art. *Computers in Industry*, 64(4):363–375.
- [Franke et al., 2011] Franke, M., Klein, P., Schröder, L., and Thoben, K.-D. (2011). Ontological semantics of standards and PLM repositories in the product development phase. In *Global Product Development*, pages 473–482. Springer.
- [Friedlander, 2009] Friedlander, J. (2009). The future of CAD 2019 as predicted by Solidworks. *New big future*.
- [Galton, 1997] Galton, A. (1997). *Spatial and temporal reasoning*. Springer Neth, oliviero stock edition.
- [Garetti, 2013] Garetti, M. (2013). Lifecycle considerations for sustainable manufacturing.
- [Gereveni and Nebel, 2002] Gereveni, A. and Nebel, B. (2002). Qualitative spatio-temporal reasoning with RCC-8 and Allen’s interval calculus: computational complexity. pages 312–316. IOS Press.
- [Gero, 1990] Gero, J. S. (1990). Design prototypes: A knowledge representation schema for design. *AI Magazine*, 11(4):26–36.
- [Gero and Kannengiesser, 2004] Gero, J. S. and Kannengiesser, U. (2004). The situated function-behaviour-structure framework. *Design Studies*, 25(4):373–391.
- [Glimm et al., 2014] Glimm, B., Horrocks, I., Motik, B., Stoilos, G., and Wang, Z. (2014). HermiT An OWL 2 reasoner. *Journal of Automated Reasoning*.
- [Gomes, 2008] Gomes, S. (2008). Ingénierie à base de connaissances pour une conception productive, optimisée, collaborative et innovante du système Projet Produit Process Usage. Habilitation à diriger les recherches, Université de Technologie de Belfort-Montbéliard.

- [Goodman, 1951] Goodman, N. (1951). *The structure of appearance*. Harvard University Press.
- [Gruber, 1993] Gruber, T. (1993). *Ontology*.
- [Guarino, 1998] Guarino, N. (1998). Formal ontology in information systems. volume 46, Italy. IOS press.
- [Haddad, 2009] Haddad, H. (2009). *Une approche pour supporter l'analyse qualitative des suites d'actions dans un environnement géographique virtuel et dynamique*. PhD thesis, Université Laval, Département d'informatique.
- [Hadjieleftheriou et al., 2002] Hadjieleftheriou, M., Kollios, G., Tsotras, V., and Gunopulos, D. (2002). Efficient indexing of spatiotemporal objects. In *Advances in Database Technology*, pages 251–268. Springer.
- [Hahmann, 2008] Hahmann, T. (2008). *Model-theoretic analysis of Asher and Vieu's mereotopology*. PhD thesis, University of Toronto.
- [Hahmann, 2012] Hahmann, T. (2012). A reconciliation of logical representations of space from multidimensional mereotopology to geometry.
- [Hales and Johnson, 2003] Hales, S. and Johnson, T. (2003). Endurantism perdurantism and special relativity. *The Philosophical Quarterly*, 53(213).
- [Haller et al., 2009] Haller, S., Karnouskos, S., and Schroth, C. (2009). The internet of things in an enterprise context. In *Future internet FIS 2008*. Domingue J., Fensel D., Traverso P., springer berlin heidelberg edition.
- [Hamilton, 2010] Hamilton, N. (2010). Managing change in manufacturing.
- [Harbelot et al., 2013] Harbelot, B., Arenas, H., and Cruz, C. (2013). The spatio-temporal semantics from a perdurantism perspective. In *Proceedings of the Fifth International Conference on Advanced Geographic Information Systems, Applications, and Services GEOProcessing*.
- [Hawley, 2004] Hawley, K. (2004). Temporal parts. *The Stanford of Encyclopedia Philosophy*.
- [Heidegger, 1962] Heidegger, M. (1962). *Being and time*. Harper and Row.
- [Helms, 2002] Helms, R. W. (2002). *Product data management as enabler for concurrent engineering*. PhD thesis, Technische Universiteit Eindhoven.
- [Hobbs and Pan, 2004] Hobbs, J. and Pan, F. (2004). An ontology of time for the semantic web. *ACM Transactions on Asian Language Information Processing*, 3(1):66–85.
- [Hoffmann and Joan-Arinyo, 2000] Hoffmann, C. and Joan-Arinyo, R. (2000). Distributed maintenance of multiple product views. *Computer-Aided Design*, 32:421–431.

- [Homem de Mello and Sanderson, 1991] Homem de Mello, L. and Sanderson, A. (1991). Representations of mechanical assembly sequences. *Robotics and Automation, IEEE Transactions*, 7(2):211–227.
- [Horrocks et al., 2012] Horrocks, I., Motik, B., and Wang, Z. (2012). The Hermit OWL Reasoner.
- [Horrocks et al., 2007] Horrocks, I., Patel-Schneider, P., McGuinness, D., and Welty, C. (2007). OWL: a description logic based ontology language for the semantic web. In *the description of logic handbook: theory, implementation and applications*. Cambridge University Press, second edition.
- [Horrocks and Sattler, 2001] Horrocks, I. and Sattler, U. (2001). Ontology in the SHOQ(D) description logic. pages 199–204, Los Altos. Morgan Kaufmann.
- [Horvath and van Der Vegte, 2003] Horváth, I. and van Der Vegte, W. F. (2003). Nucleus-Based Product Conceptualization-Part 1: Principles and Formalization. In *International Conference on Engineering Design*.
- [Horvath et al., 1998] Horváth, I., Vergeest, J., and Kuczogi, G. (1998). Proceedings of ASME DETC. In *Development and application of design concept ontologies for contextual conceptualization*, pages 1–16.
- [Huang et al., 2001] Huang, G., Yee, W., and Mak, K. (2001). Development of a web based system for engineering change management. *Robotics and Computer Integrated Manufacturing*, 17(3):255–267.
- [Issa et al., 2003] Issa, R., Flood, I., and O’Brien, W. (2003). *4D CAD and visualization in construction Developments and Applications*. Taylor and Francis.
- [Kim et al., 2006] 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.
- [Kim et al., 2004] 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.
- [Kim et al., 2008] 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(6):744–754.
- [Kiritsis, 2011] Kiritsis, D. (2011). Closed-loop PLM for intelligent products in the era of the Internet of things. *Computer-Aided Design*, 43(5):479–501.
- [Kiritsis et al., 2003] Kiritsis, D., Bufardi, A., and Xirouchakis, P. (2003). Research issues on product lifecycle management and information tracking using smart embedded systems. *Advanced Engineering Informatics*, 17(3-4):189–202.



- [Kitamura et al., 2004] Kitamura, Y., Kashiwase, M., Fuse, M., and Mizoguchi, R. (2004). Deployment of an ontological framework of functional design knowledge. *Advanced Engineering Informatics*, 18(2):115–127.
- [Koufteros et al., 2001] Koufteros, X., Vonderembse, M., and Doll, W. (2001). Concurrent engineering and its consequence. *Journal of Operations Management*, 19(1):97–115.
- [Kubler et al., 2015] Kubler, S., Derigent, W., Framling, K., Thomas, A., and Rondeau, E. (2015). Enhanced Product Lifecycle Information Management using communicating materials. *Computer-Aided Design*, 59:192–200.
- [Kusiak, 2006] Kusiak, A. (2006). Data mining: manufacturing and service applications. *International Journal of Production Research*, 44(18-19):4175–4191.
- [Kusiak and Salustri, 2007] Kusiak, A. and Salustri, F. A. (2007). Computational intelligence in product design engineering: review and trends. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, 37(5):766–778.
- [Labrousse and Bernard, 2008] Labrousse, M. and Bernard A. (2008). *FBS-PPRE, an Enterprise knowledge lifecycle model*. Methods and tools for effective knowledge lifecycle-management, Springer.
- [Laroche, 2007] Laroche, F. (2007). *Contribution à la sauvegarde des objets techniques anciens par l’archéologie industrielle avancée. Proposition d’un modèle d’information de référence muséologique et d’une méthode inter-disciplinaire pour la capitalisation des connaissances du patrimoine technique et industriel*. PhD thesis, Ecole centrale de Nantes.
- [Le Moigne, 1994] Le Moigne, J.-L. (1994). *La théorie du système général : théorie de la modélisation*. Presses universitaires de France.
- [Lee and Jeong, 2012] Lee, J. and Jeong, Y. (2012). User centric knowledge representations based on ontology for AEC design collaboration. *Computer-Aided Design*, pages 735–748.
- [Lemon and Pratt, 1998] Lemon, O. and Pratt, I. (1998). Complete logics for QSR: a guide to plane mereotopology. *Journal of Visual Languages and Computing*, 9:5–21.
- [Leonard and Goodman, 1940] Leonard, H. and Goodman, N. (1940). The calculus of individuals and its use. *Journal of Symbolic Logic*, 5(2):45–55.
- [Lesniewski, 1929a] Lesniewski, S. (1929a). Fundamentals of a new system of the foundations of mathematics. *Fundamenta Mathematicae*, 14:1–81.
- [Lesniewski, 1929b] Lesniewski, S. (1929b). On the foundations of ontology. *Comptes rendus des séances de la société des sciences et des lettres de Varsovie*, 22:111–132.
- [Levenchuck, 2012] Levenchuck, A. (2012). Ontology engineering for systems engineering.

- [Liu and Sun, 2008] Liu, J. and Sun, Z. (2008). *Representing design intents for design thinking process modelling*. Yan, X.-T., Lon, W.J., Eynard, B.
- [Louhichi and Rivest, 2014] Louhichi, B. and Rivest, L. (2014). Maintaining consistency between CAD elements in collaborative design using association management and propagation. *Computers in Industry*, 65:124–135.
- [Lu et al., 2009] Lu, M., Zhang, Y., Zhang, J., Hu, Z., and Li, J. (2009). Integration of four-dimensional computer-aided design modeling and three-dimensional animation of operations simulation for visualizing construction of the main stadium for the Beijing 2008 Olympic games. *Revue canadienne de génie civil*, 36(3):473–479.
- [MacKrell, 2015] MacKrell, J. (2015). Managing product changes.
- [Matsokis and Kiritsis, 2010] Matsokis, A. and Kiritsis, D. (2010). An ontology-based approach for Product Lifecycle Management. *Computers in Industry*, 61(8):787–797.
- [McKinney et al., 1996] McKinney, K., Kim, J., Fischer, M., and Howard, C. (1996). Interactive 4d-CAD. In *Proceedings of the third Congress on Computing in Civil Engineering*, pages 383–389.
- [Mignard and Nicholle, 2014] Mignard, C. and Nicholle, C. (2014). Merging BIM and GIS using ontologies application to urban facility management in ACTIe3d. *Computers in Industry*, 65:1276–1290.
- [Monticolo, 2008] Monticolo, D. (2008). *Une approche organisationnelle pour la conception d’un système de gestion des connaissances fondé sur le paradigme agent*. PhD thesis, Université de Technologie de Belfort-Montbéliard.
- [Morbach, 2009] Morbach, J. (2009). *A reusable ontology for computer-aided process engineering*. PhD thesis, Fakultät für Maschinenwesen der RWTH Aachen.
- [Muller, 1998] Muller, P. (1998). A qualitative theory of motion based on spatio-temporal primitives. In *Principles of Knowledge Representation and Reasoning-International Conference*, pages 131–143.
- [Muller, 2002] Muller, P. (2002). Topological spatio-temporal reasoning and representation. *Computational Intelligence*, 18(3):420–450.
- [Nonaka and von Krogh, 2009] Nonaka, I. and von Krogh, G. (2009). Tacit Knowledge and Knowledge Conversion: Controversy and Advancement in Organizational Knowledge Creation Theory. *Organization Science*, 20(3):635–652.
- [Parametric Technology Corporation, 2006] Parametric Technology Corporation (2006). Support PTC : Gestion des modifications et des configurations.
- [Parent et al., 2006] Parent, C., Spaccapietra, S., and Zimanyi, E. (2006). *Conceptual modeling for traditional and spatio-temporal applications: the MADS approach*. Springer.



- [Peachavanish et al., 2006] Peachavanish, R., Karimi, H. A., Akinci, B., and Boukamp, F. (2006). An ontological engineering approach for integrating CAD and GIS in support of infrastructure management. *Advanced Engineering Informatics*, 20(1):71–88.
- [Pease, 2014] Pease, A. (2014). Suggested Upper Merged Ontology (SUMO).
- [Pikosz and Malmqvist, 1998] Pikosz, P. and Malmqvist, J. (1998). A comparative study of engineering change management in three Swedish engineering companies. In *Proceedings of the DETC98 ASME design engineering technical conference*, pages 78–85.
- [Pittet et al., 2014] Pittet, P., Cruz, C., and Nicholle, C. (2014). An ontology change management approach for facility management. *Computers in Industry*, 65:1301–1315.
- [PLM IWG, 2007] PLM IWG (2007). A new point of view on PLM. White paper of the International Working Group on PLM.
- [Pooyan et al., 2009] Pooyan, A., Griffis, F., and Lawrence, C. (2009). Building information model the role and need of the constructors building a sustainable future. pages 467–476.
- [Pourtalebi et al., 2014] Pourtalebi, S., Horváth, I., and Opiyo, E. (2014). First steps towards a mereo-operandi theory for a system feature-based architecting of cyber-physical systems. Germany.
- [Pratt and Schoop, 2000] Pratt, I. and Schoop, D. (2000). Expressivity in polygonal, plane mereotopology. *Journal of Symbolic Logic*, 65:822–838.
- [Quintana et al., 2012] Quintana, V., Rivest, L., Pellerin, R., and Kheddouci, F. (2012). Re-engineering the engineering change management process for a drawing-less environment. *Computers in Industry*, 63:79–90.
- [Rahmani and Thomson, 2012] Rahmani, K. and Thomson, V. (2012). Ontology based interface design and control methodology for collaborative product development. *Computer-Aided Design*, 44(5):432–444.
- [Randell et al., 2012] Randell, D., Landini, G., and Galton, A. (2012). Discrete Mereotopology for Spatial Reasoning in Automated Histological Image Analysis.
- [Randell and Cohn, 1989] Randell, D. A. and Cohn, A. G. (1989). Exploring naive topology: Modelling the force pump. In *Proceedings of the Third International Qualitative Physics Workshop*.
- [Randell et al., 1992] Randell, D. A., Cui, Z., and Cohn, A. G. (1992). A spatial logic based on regions and connection. 92:165–176.
- [Renolen, 1999] Renolen, A. (1999). Concepts and methods for modelling temporal and spatiotemporal information. Technical report, NTNU.

- [Rezgui et al., 2011] Rezgui, Y., Boddy, S., Wetherill, M., and Cooper, G. (2011). Past present future of information and knowledge sharing in the construction industry towards semantic service based e construction. *Computer-Aided Design*, 43:502–215.
- [Riemenschneider, 2014] Riemenschneider, R. (2014). Work programme 2014-2015 ICT for factories of the future.
- [Robinson, 2012] Robinson, M. A. (2012). How design engineers spend their time: Job content and task satisfaction. *Design Studies*.
- [Rodier et al., 2010] Rodier, X., Saligny, L., Lefebvre, B., and Pouliot, J. (2010). ToToPI a GIS for understanding urban dynamics based on the OH FET model. pages 337–349, Virginia, USA. Fricher B., Crawford J., Koler D.
- [Ross et al., 2008] Ross, A. M., Rhodes, D. H., and Hastings, D. E. (2008). Defining changeability Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. *Systems Engineering*, 11(3):246–262.
- [Rouibah and Caskey, 2003] Rouibah, K. and Caskey, K. (2003). Change management in concurrent engineering from a parameter perspective. *Computers in Industry*, 50:15–34.
- [Rusak et al., 2004] Rusak, Z., Horváth, I., and Van der Vegte, W. (2004). First step towards an all embracing relations based modeling. Salt Lake City, Utah.
- [Salustri et al., 2008] Salustri, F., Eng, N. L., and Weerasinghe, J. S. (2008). Visualizing information in the early stages of engineering design. *Computer-Aided Design and Applications*, 5(5):697–714.
- [Salustri, 2000] Salustri, F. A. (2000). Towards a Logical Framework for Engineering Design Processes. In *From Knowledge Intensive CAD to Knowledge Intensive Engineering*, volume 207, pages 201–212. Umberto Cugini and Michael J. Wozny, ifip conference proceedings edition.
- [Salustri, 2002] Salustri, F. A. (2002). Mereotopology for product modeling. A new framework for product modeling based on logic. *The Journal of Design Research*, 2.
- [Salustri, 2012] Salustri, F. A. (2012). Representing CK theory with an action logic.
- [Salustri and Lockledge, 1999] Salustri, F. A. and Lockledge, J. C. (1999). Towards a formal theory of products including mereology. In *Proceeding of 12th International Conference on Engineering Design*, pages 1125–1130.
- [Santocchi and Dini, 1992] Santocchi, M. and Dini, G. (1992). Computer-aided planning of assembly operations: the selection of assembly sequences. *Robotics and Computer Integrated Manufacturing*, 9(6):439–446.
- [Sartre, 1946] Sartre, J.-P. (1946). *L’existentialisme est un humanisme*. Nagel Paris.

- [Schockaert et al., 2007] Schockaert, S., De Cock, M., and Kerre, E. (2007). Qualitative temporal reasoning about vague events. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 569–574.
- [Shadbolt et al., 2006] Shadbolt, N., Hall, W., and Berners-Lee, T. (2006). The semantic web revisited. *Intelligent Systems, IEEE*, 21(3):96–101.
- [Sider, 2001] Sider, T. (2001). *Four dimensionalism: an ontology of persistence and time*. Oxford, Clarendon Press.
- [Siemens Industry Software, 2015] Siemens Industry Software (2015). PDM Gestion des données relatives aux produits.
- [Singh et al., 2009] Singh, V., Skiles, S. M., Krager, J. E., Wood, K. L., Jensen, D., and Sierakowski, R. (2009). Innovations in Design Through Transformation: A Fundamental Study of Transformation Principles. *Journal of Mechanical Design*, 131(8).
- [Singh et al., 2007] Singh, V., Walther, B., Krager, J., Putnam, N., Koraisly, B., Wood, K. L., and Jensen, D. (2007). Design for transformation: Theory, method and application. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 447–459. American Society of Mechanical Engineers.
- [Smith, 1996] Smith, B. (1996). Mereotopology: a theory of parts and boundaries. *Data and Knowledge Engineering*, 20(3):287–303.
- [Smith and Eppinger, 1997] Smith, R. and Eppinger, S. (1997). Identifying controlling features of engineering design iteration. *Management Science*, (43):276–293.
- [Standard ISO 15926, 2003] Standard ISO 15926 (2003). Integration of lifecycle data for process plant including oil and gas production facilities.
- [Stark, 2006] Stark, J. (2006). *Product Lifecycle Management. 21st century paradigm for product realization*. Berlin.
- [Stefani et al., 2009] Stefani, C., De Luca, L., and Véron, P. (2009). Time indeterminacy and spatio-temporal building transformations: an approach for architectural heritage understanding. *International Journal of Interactive Design Manufacturing*, 4:61–74.
- [Stell and West, 2004] Stell, J. G. and West, M. (2004). A four-dimensionalist mereotopology. In *Formal Ontology in Information Systems*, pages 261–272.
- [Strange et al., 2014] Strange, H., Chen, Z., Denton, E. R., and Zwiggelaar, R. (2014). Modelling mammographic microcalcification clusters using persistent mereotopology. *Pattern Recognition Letters*, 47:157–163.
- [Straume, 2014] Straume, K. (2014). The social construction of a land cover map and its implications for Geographical Information Systems (GIS) as a management tool. *Land Use Policy*, 39:44–53.

- [Sun et al., 2010] Sun, W., Ma, Q.-Y., Gao, T.-Y., and Chen, S. (2010). Knowledge-intensive support for product design with an ontology-based approach. *The International Journal of Advanced Manufacturing Technology*, 48(5-8):421–434.
- [Swain et al., 2014] Swain, A., Sen, D., and Gurumoorthy, B. (2014). Extended liaison as an interface between product and process model in assembly. *Robotics and Computer-Integrated Manufacturing*, 30(5):527–545.
- [Tarski, 1956] Tarski, A. (1956). *Foundations of the geometry of solids*. Trans. Hackett, woodger, j., corcoran, j. edition.
- [Tavcar and Duhovnik, 2005] Tavcar, J. and Duhovnik, J. (2005). Engineering change management in individual and mass production. *Robotics and Computer-Integrated Manufacturing*, 21:205–215.
- [Tudorache, 2006] Tudorache, T. (2006). *Employing Ontologies for an Improved Development Process in Collaborative Engineering*. PhD thesis, Universitätsbibliothek.
- [Uschold and Gruninger, 1996] Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11.
- [Valle and Vazquez-Bustelo, 2009] Valle, S. and Vazquez-Bustelo, D. (2009). Concurrent engineering performance: Incremental versus radical innovation. *International Journal of Production Economics*, 119(1):136–148.
- [Van der Vegte and Horvath, 2003] Van der Vegte, W. F. and Horvath, I. (2003). Nucleus-based product conceptualization: application in designing for use. In *ICED*.
- [Van der Vegte et al., 2008] Van der Vegte, W. F., Horvath, I., and Rusak, Z. (2008). Simulating the use of products: applying the nucleus paradigm to resource-integrated virtual interaction models. In *Proceedings of the 7th International Symposium on Tools and Methods of Competitive Engineering (TMCE'08)*, volume 1, pages 591–605.
- [Van Harmelen et al., 2007] Van Harmelen, F., Lifschitz, V., and Porter, B. (2007). *Handbook of knowledge representation*. Foundations of Artificial Intelligence. Elsevier science edition.
- [Varzi, 1996] Varzi, A. C. (1996). Parts, wholes, and part-whole relations: The prospects of mereotopology. *Data and Knowledge Engineering*, 20(3):259–286.
- [Varzi, 1998] Varzi, A. C. (1998). Basic problems of mereotopology. In *Formal Ontology in Information Systems*. IOS Press, pages 29–38, Italy. IOS Press.
- [Viorel, 2007] Viorel, M. (2007). *TOWL Language requirements*.
- [Wang et al., 2012] Wang, H., Johnson, A., and Bracewell, R. (2012). The retrieval of structured design rationale for the re-use of design knowledge with an integrated representation. *Advanced Engineering Informatics*, 26:251–266.





## Résumé :

L'objectif de cette thèse est de décrire l'évolution du produit dans les trois dimensions (spatiale, temporelle et spatio-temporelle). Dans le contexte industriel actuel, les modèles produit sont considérés uniquement du point de vue spatial pendant la phase de conception et du point de vue temporel pendant la phase d'assemblage. Le manque de lien entre le produit et le process mène à des incompréhensions de définition de produit et entraîne de mauvaises interprétations en conception. Cependant, le produit évolue à travers le temps et subit des changements tout au long des phases de conception et d'assemblage. L'aspect dynamique des activités de conception nécessite de lier ces deux dimensions afin de pouvoir représenter l'évolution du produit et avoir une cohérence des informations. Par conséquent, la dimension spatio-temporelle (i.e. permettant de lier l'espace et le temps) a besoin d'être ajoutée et les relations entre la modélisation du produit et sa séquence d'assemblage ont besoin d'être particulièrement étudiées.

Cette thèse en mécanique et conception s'est inspirée de divers domaines comme la gestion des connaissances, les systèmes d'information géographique et la philosophie. Ici le produit est considéré d'un point de vue perdurantiste. Le perdurantisme considère l'objet comme étant composé de tranches temporelles et gardant toujours la même identité quelque soit le changement subi. D'après les précédentes déclarations, cette thèse introduit une nouvelle description du couple produit-process afin d'assurer la compréhension des intentions de conception aux acteurs projet. Dans le but d'atteindre cet objectif, une théorie méréotopologique, permettant de décrire le produit comme perçu dans la réalité, a été développée et implémentée dans un modèle ontologique pour être formalisée.

La théorie JANUS décrit qualitativement l'évolution du produit à travers le temps dans un contexte de conception orientée assemblage, permettant l'intégration de la séquence d'assemblage dès le début du processus de conception. La théorie permet la description formelle des relations liant les informations et connaissances du couple produit-process. Ces efforts ont pour but d'apporter une base concrète pour la description des changements d'entités spatiales (telles que les composants) et leurs relations à travers l'espace et le temps. Cette théorie basée sur les régions lie les dimensions spatiale, temporelle et spatio-temporelle et apporte donc une vision perdurantiste en conception de produit.

Ensuite, PRONOA2 – une ontologie formelle basée sur la précédente théorie – a été développée. De ce fait, les informations liées à l'assemblage sont rendues accessibles et exploitables par des systèmes de gestion d'information et les outils de XAO afin de supporter les activités de l'architecte produit et du concepteur. En effet, les informations et connaissances liées à la conception de produit, ainsi que la séquence d'assemblage associée, ont besoin d'une fondation sémantique et logique afin d'être gérées de manière cohérente et proactive.

Suite au développement de la théorie JANUS et de l'ontologie PRONOA2, l'approche proposée permet d'associer les informations spatiales (gérées par le PDM) et les informations temporelles (gérées par le MPM) à travers des relations méréotopologiques spatio-temporelles. Par conséquent, de nouvelles entités doivent être gérées dans le PLM, en utilisant notamment l'ontologie et un système hub, afin d'assurer un maintien des principes d'ingénierie proactives et améliorer la compréhension de l'architecte produit et du concepteur concernant l'évolution du produit.

**Mots-clés :** Gestion du cycle de vie du produit, Conception-orientée assemblage, Méréotopologie, Description qualitative du couple produit-process, Spatio-temporel, Ontologie

## Abstract:

The major goal of this research is to describe product evolution in the three dimensions (i.e. spatial, temporal and spatiotemporal). In the current industrial context, product models are only considered from a purely spatial point of view during the design stage and from a purely temporal point of view during the assembly stage. The lack of link between product and process leads to misunderstanding in engineering definition and causes wrong design interpretation. However, the product undergoes changes throughout the design and assembly phases. The dynamic aspect of design activities requires linking both dimensions in order to be able to represent product evolution and have consistent information. As such, spatiotemporal dimension (i.e. linking space and time) needs to be added and relationships between product modelling and assembly sequences need to be particularly studied.

This PhD thesis in mechanical design draws inspiration from several domains such as mathematics, geographic information systems and philosophy. Here the product is considered from a perdurantist point of view. Perdurantism regards the object as being composed of temporal slices and always keeping the same identity whatever changes undergone. Based on this statement, this PhD thesis introduces a novel product-process description so as to ensure product architect's and designer's understanding of design intents at the early design stages. In order to achieve this objective, a mereotopological theory, enabling the product description as it is perceived in the real world, has been developed and implemented in an ontology model to be formalized.

The JANUS theory qualitatively describes product evolution over time in the context of AOD, integrating assembly sequence planning in the early product design stages. The theory enables the formal relationships description of product-process design information and knowledge. The proposed efforts aim at providing a concrete basis for describing changes of spatial entities (i.e. product parts) and their relationships over time and space. This region-based theory links together spatial, temporal and spatiotemporal dimensions, therefore leading to a perdurantist philosophy in product design.

Then, PRONOA2 – a formal ontology based on the previous mereotopological theory – is developed. Assembly information is accessible and exploitable by information management systems and computer-aided X tools in order to support product architects and designer's activities. Indeed product design information and knowledge as well as the related assembly sequence require a semantic and logical foundation in order to be managed consistently and processed proactively.

Based on JANUS theory and PRONOA2 ontology, the proposed approach enables associating spatial information (managed by PDM) and temporal information (managed by MPM) through spatiotemporal mereotopological relationships. Therefore, new entities are managed through PLM, using ontology and hub system, so as to ensure proactive engineering and improve product architects' and designers' understanding of product evolution.

**Keywords:** Product Lifecycle Management, Assembly-Oriented Design, Mereotopology, Product-Process Qualitative Description, Spatiotemporal, Ontology

■ École doctorale SPIM - Université de Technologie Belfort-Montbéliard

F - 90010 Belfort Cedex ■ tél. +33 [0]3 84 58 31 39

■ ed-spim@univ-fcomte.fr ■ www.ed-spim.univ-fcomte.fr

