



HAL
open science

Empty Space Modelling for Detecting Spatial Conflicts Across Multiple Design Domains

Arun Kumar Singh, B. Gurumoorthy, Latha Christie

► **To cite this version:**

Arun Kumar Singh, B. Gurumoorthy, Latha Christie. Empty Space Modelling for Detecting Spatial Conflicts Across Multiple Design Domains. 14th IFIP International Conference on Product Lifecycle Management (PLM), Jul 2017, Seville, Spain. pp.223-230, 10.1007/978-3-319-72905-3_20. hal-01764187

HAL Id: hal-01764187

<https://inria.hal.science/hal-01764187>

Submitted on 11 Apr 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.



Distributed under a Creative Commons Attribution 4.0 International License

Empty space modelling for detecting spatial conflicts across multiple design domains

Arun Kumar Singh^{1,2}, B. Gurumoorthy², Latha Christie¹

¹ Microwave Tube Research and Development Centre (DRDO), Bangalore- 560013, India

² Centre for Product Design and Manufacturing, Indian Institute of Science, Bangalore - 560012, India

Abstract. This paper discusses identification and decomposition of empty spaces in Computer Aided Design (CAD) model for detecting spatial conflicts across multiple design domains. An Algorithm has been developed to identify empty spaces in CAD model and decompose it to a level, where it can be correctly associated with the connected domains, states and requirements. Knowledge capture and representation have been demonstrated with System Modelling Language (SysML) diagrams using SysML tools. SysML blocks have been introduced to define intended empty spaces, product states and design domain in SysML structure diagrams. Association of these blocks with neighboring parts, has also been discussed. A case study of heat sink assembly has been taken for Empty space modelling.

Keywords: SysML, Empty Space modelling, Spatial conflict

1 Introduction

Product designs are carried out with objectives to satisfy multifunctional domain requirements. In every product design, some spaces are intentionally left empty to satisfy specific requirements of different domains. For example space may have been provided for electrical insulation, thermal isolation, clearance for moving part, port accessibility, Electron beam or Radio frequency interaction, expansion of material, tool access, sensors, bolts accessibility, welding tools, or for removal / replacement of parts. There may be cases, where the same space is used for different purposes but in different product states. The space, which has been provided for moving parts clearance in operation state also serves the purpose for accessibility in maintenance state. That is the same space, which is used for two different purposes but in different states. Knowledge regarding these space usage for moving part clearance under operation state, may not have been documented or is not available to designer, who is searching for empty space to locate new module into the product. This may create spatial conflict with workspace of other designer in the same domain or with other domain designers. This also incurs time wastage; if these spatial conflicts are detected and resolved through conventional methods of multi-domain design stakeholder reviews.

Electro-Mechanical systems exhibit more examples of these types of conflict. Designers from different domains have spatial conflicts, who claim over same empty space in coupled design iterations at different timelines.

CAD tools only hold the geometry information of the product, while beyond geometry information is not captured in CAD models. Product design information like requirements, product states, and actors involved at different product states, functional model and behaviour model, is not available in CAD model. Use of tools like System Modelling Language (SysML) by incorporation of system engineering domain, provides a way to capture and to represent product's non-geometry design knowledge. Problem at hand needs identification of intended empty spaces and its decomposition to correctly associate a portion of this empty space with the corresponding non-geometry information.

This problem for spatial conflict detection can be divided mainly into three levels: first level is identification of bulk of Empty space available in products, decomposition of this bulk empty space block into smaller empty space blocks and incorporation of non-geometry product knowledge by modelling it as system level domain information in SysML. Second level is building associativity among these empty spaces, product parts and SysML Information. The third level is spatial conflict detection using first two levels of information. In this paper, first level has been attempted.

2 System Modelling Language (SysML) Model and Diagrams

Before identification of empty spaces in product, methods are needed for capturing and representing non-geometry product knowledge. There are various product models available to capture non-geometry product information like NIST Core Product Model (CPM) [1-2], its extension Open Assembly Model (OAM) [3] and Methodology and tools Oriented to Knowledge-based engineering Applications (MOKA) [4-5]. These product models have classes to define product form, assembly structure, function, behavior and technology. These models are developed in Unified Modelling Language (UML). These product models do not have classes to define empty spaces and its associations with product knowledge. In recent times, more studies have been done to improve UML for incorporating system engineering domain information. SysML [6] is the extension of UML, which is developed to provide standard system modelling tools. SysML supports capturing requirements of system and capturing structural & behavioral models. SysML also supports equation based behavior through constraint blocks. Block Definition Diagrams (BDD) and Internal Block Diagrams (IBD) are used to define classes and instances. BDD and IBD are extended from UML structure diagram. SysML contains two new diagrams namely requirement diagram and Parametric Diagram, which were not present in UML2 diagrams.

Use of SysML tools, provides the way to capture, represent and associate the product's non-geometry information. There are no direct methods in SysML to resolve spatial conflicts of product design, as it is only a modelling language. Higher modelling efforts are needed to model information manually in SysML for any product.

Hence automation of knowledge capture should be a criteria to generate information wherever possible by interfaces and standards. Bringing information in structured form inside SysML, can be a step towards interoperability as information is available in open standards and in structured form like XML. Having information in system engineering domain brings new challenges of maintaining consistency among developing models. Different designers use different design tools which do not communicate directly and have been proposed to be integrated by common framework of system engineering domain [7].

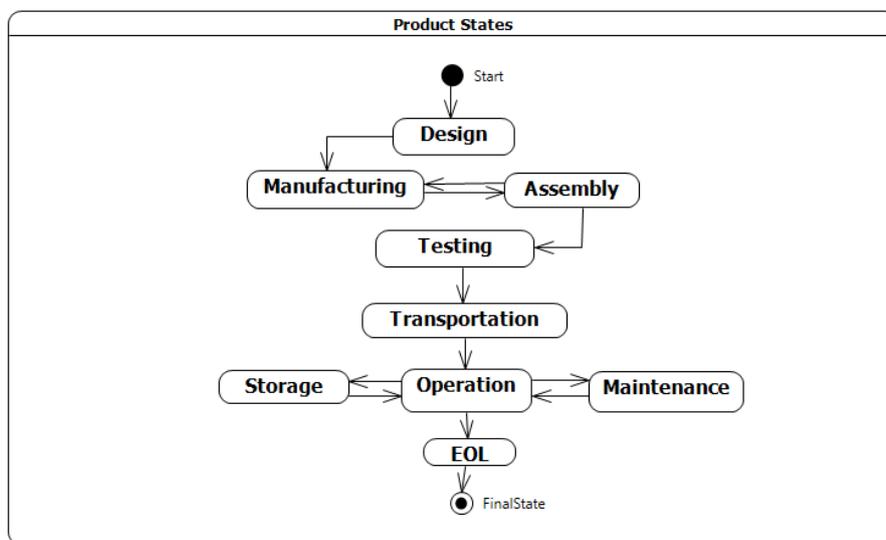


Fig. 1. Typical Product States and Transitions in a Product Lifecycle

It is intended to detect spatial conflicts at different timelines of product lifecycle. In Building Information Management (BIM) tools, 4D modeling is used to detect such spatial conflicts [8]. It is like 3D modelling with an additional time variable. In product lifecycle, the time variable can be considered as product states. Product goes through manufacturing and assembly after design completion. Then it may go to transportation state before getting commissioned for operation. It may require periodic maintenance. A typical SysML product state diagram has been shown in fig 1. Product Design may involve designers from different domains, using different tools and working on product at different timelines. There can be domains involved in product design like structural, thermal, manufacturing, ergonomics, electrical, electromagnetic etc.

CAD tools hold product geometry information so the same CAD tools should be holding the geometry of the empty spaces. Empty space block geometry data can also be saved in open standards like STEP, if multi-domain tools have to access this information. Other information for these blocks should be inside the SysML Product Model. SysML blocks have been introduced to define intended empty spaces. Blocks

(Classes) for Empty space, product states and design domain, have been added in SysML Block Definition Diagram (BDD) shown in fig 2. An assembly is composed of parts and empty space blocks. An assembly can also aggregate other subassemblies. This has been shown in fig 2 by having a self-aggregation. Empty space blocks have direct associations with parts, product states and design domains with a multiplicity of (0 *) as shown in fig 2. These blocks can have inferred associations with requirements and functional diagram through Parts, states or domains.

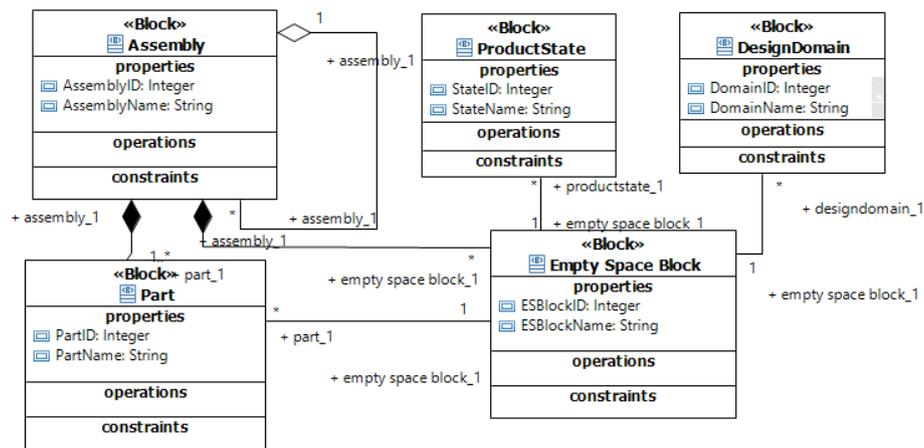


Fig. 2. SysML Block Definition Diagram (BDD) with Empty Space Block

3 Identification and Decomposition of Empty Spaces

The first step in spatial conflict detection is identification of empty spaces available to be used at that design state. CAD tools are used for 3D modelling of product components and assemblies. There are different levels of sub-assemblies used to make the design modular in nature. Empty spaces available for modelling at part level are different when same parts are assembled together to from an assembly. Identification of these spaces then becomes an assembly level affair than to find it on part level.

Code has been generated to identify and decompose empty spaces for assemblies in SolidWorks with Visual Basic Application (VBA) API. These are the steps used in code for identification and decomposition of Empty Spaces

- Extract solid bodies from the Product Assembly into a new part.
- Calculate Bounding Box Dimensions surrounding these solid bodies.
- Create a Solid Block with the bounding box dimensions not merging it with the existing solid bodies.
- Subtract Product solid bodies from the bounding box main block.
- Extract planes/surfaces derived from selected faces from Product Assembly.

- Split this empty space block into a number of solid bodies using extracted planes.
- Save these blocks into a separate folder as separate entities. Convert these parts into assembly feature in SolidWorks to make every empty space block editable and to have mating features based on the part associations.

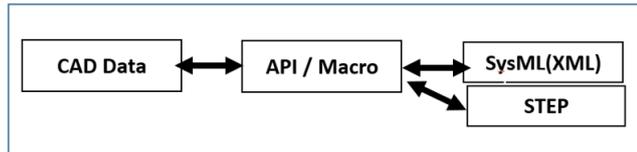


Fig. 3. Schematic showing code interaction with CAD and SysML

SysML Diagrams have been generated using open source code Papyrus on Eclipse IDE. Fig 3 shows the schematic of envisaged code interaction with CAD model and SysML model to fully automate the task of knowledge capture. SysML data is held in XML format, which makes it easy to expose and interface it with other applications and tools. A commercial CAD system SolidWorks is being used, only for accessing geometry. STEP has been chosen to store the product geometry data for the proposed implementation as that makes the solution vendor neutral.

4 Case Study

A heat Sink assembly, which is used to cool electronic circuits has been taken as a case study for empty space identification and decomposition. SysML Requirement diagram has been shown for Heat Sink assembly in fig 4. Heat sink requirements are composed of electronic circuit requirements, thermal design requirements and manufacturing feasibility shown in diagram. Maximum allowable temperature limits are decided by the Electronic circuit component datasheets, which falls into the electronic domain. Thermal designer does fan selection and designs heat sink parameters like fin length, depth, gap and width. Fan flow rates and pressure drops are associated with fan selection. Final pressure drops depend upon heat sink fin parameters like fin depth and fin gap. The same parameters are driven by the manufacturing domain as these have to be fabricated and the cutting tool should be available to machine these fins gaps with that fin depth. Complexity of associativity and design dependency for a simple heat sink design, have been depicted using this SysML requirement diagram.

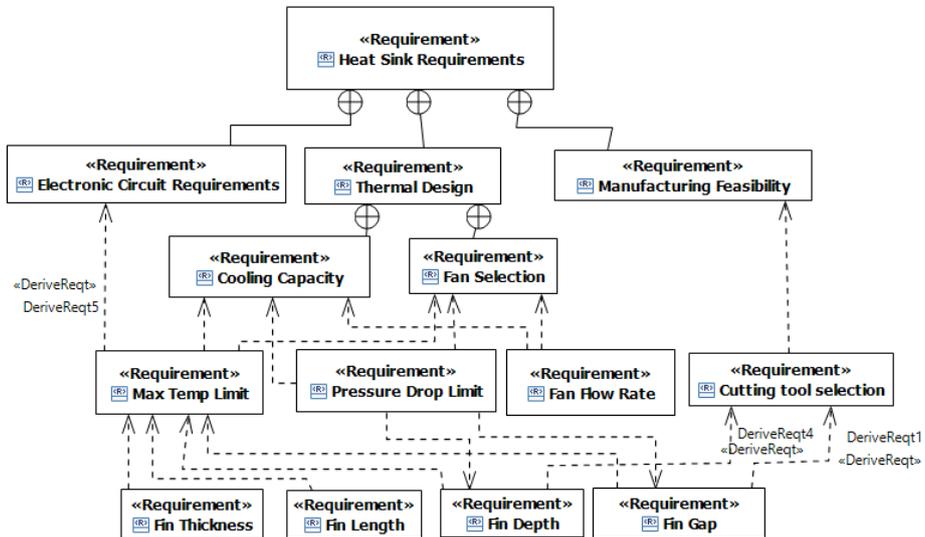


Fig. 4. SysML requirement diagram for Heat Sink Assembly

3D Model for Heat sink assembly has been generated in SolidWorks and is shown in fig 5a. Bounding box coordinates are calculated and a Bounding box solid body is created with these coordinates as shown in fig 5b. This box has been used as main block and remaining heat sink assembly has been subtracted from it. This results into the part shown in fig 5c, which are empty spaces available in Heat Sink Assembly. As seen there are counter bore holes made for screws needed to assemble heat sink. But other than these, a bulk of space is available as single part, which needs to be further decomposed.

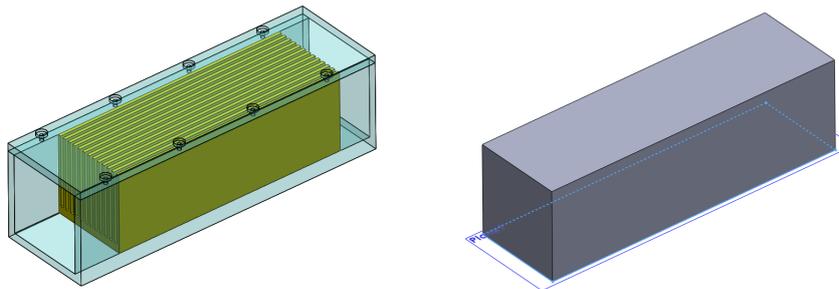


Fig. 5. a) 3D model of Heat Sink Assembly. b) Bounding Box

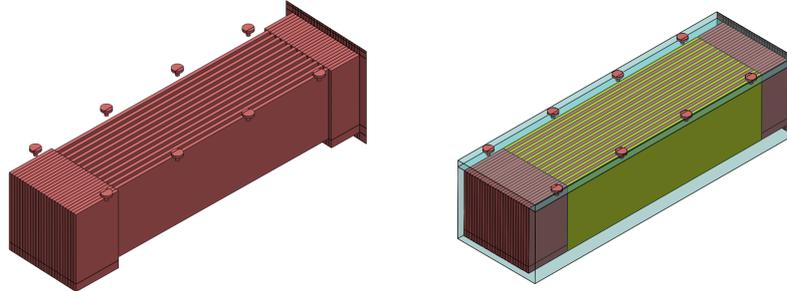


Fig. 5. c) Identified and decomposed Empty spaces. d) Empty space solid bodies assembled back with the Heat sink assembly.

Face selections are made to split the bulk volumes into smaller blocks, which are used to decompose this block into smaller blocks. This empty space model has been inserted back into heat sink assembly and shown in fig 5d. Decomposition is needed to such a level that can correctly identify and represent air flow between two fins with a decomposed block. Any obstruction made in-between heat sink fins, restricts air flow under operation and in turn reduces heat sink cooling capacity of device. Any part located in airflow path should be a conflict with thermal domain.

All these empty space blocks can be captured in SysML IBD with block definitions, which are done in BDD as shown in fig 2. At present IBD has been created using SysML Tool Papyrus as shown in fig 6. Heat Sink, Enclosure and Decomposed Empty space blocks are modelled as instances inside heat sink assembly IBD. Connections have been shown to represent associations among them. Generation of IBD and connections have been envisaged to be done automatically in future.

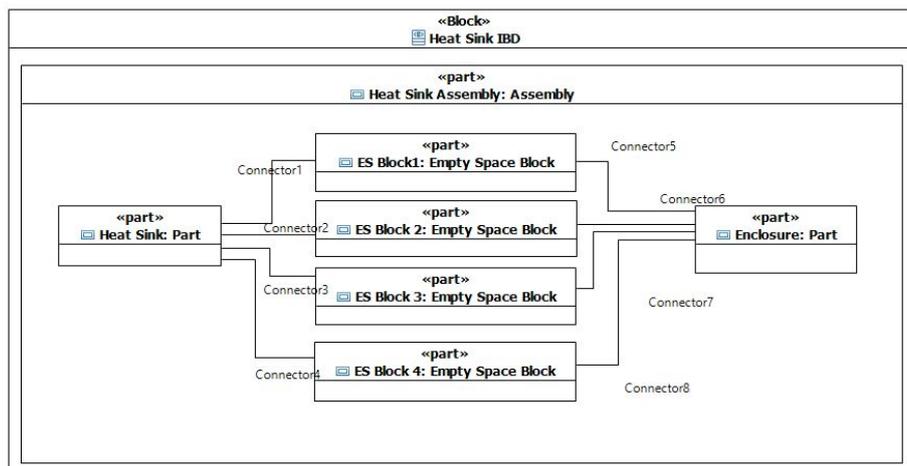


Fig. 6. Internal Block Diagram (IBD) for Heat Sink Assembly showing only selective empty space block instances and association with parts.

5 Conclusion

New blocks have been defined in SysML model for empty spaces, parts, assemblies, product states and domains. These blocks can be used to capture and represent knowledge inside CAD tools. Identification and decomposition of empty spaces inside CAD have been discussed and code has been generated for automation task. SysML diagram has been generated in SysML tools to model CAD assembly structure and generated empty space blocks.

Association of this knowledge with Empty space blocks inside CAD tools has not been done now and has been proposed for future work. Spatial Conflict detection, which needs first two level of information of Empty space blocks, SysML information and their associations, is proposed for future work.

6 Acknowledgement

The First author is thankful to Director, Microwave Tube Research & Development Centre (MTRDC) for providing the opportunity and encouragement in carrying out the work presented in the paper.

7 References

1. Steve J. Fenves, Sebti Foufou, Conrad Bock, Ram D. Sriram (2008) CPM2: A Core Model for Product Data, *Journal of Computing and Information Science in Engineering*, ASME, 8 (1), 2008.
2. Sebti Foufou, Steve J. Fenves, Conrad Bock, Rachuri Sudarsan, Ram Sriram (2005) A Core Product Model for PLM with an illustrative XML implementation, *Proceedings of the PLM International Conference on Product Lifecycle Management*, Inderscience Publishers, Lyon, France, pp. 21-32, July 2005.
3. Sudarsan Rachuri, Young-Hyun Han, Sebti Foufou, Shaw C. Feng, Utpal Roy, Fujun Wang, Ram D. Sriram, Kevin W. Lyons (2006) A Model for Capturing Product Assembly Information, *Journal of Computing and Information Science in Engineering*, 6(1), p. 11-21, 2006.
4. Sudarsan, R., Young-Hyun H, Feng, S. C., Roy, U., Fujun W., Sriram, R. D., and Lyons, K. W., (2003) Object-oriented Representation of Electro-Mechanical Assemblies Using UML, National Institute of Standards and Technology, NISTIR 7057, Gaithersburg, MD 20899, USA, 2003.
5. Oldham K, Kneebone S, Callot M, Murton A, Brimble R. (1998) MOKA – A Methodology and tools Oriented to Knowledge-based engineering Applications. In: *Proceedings of the Conference on Integration in Manufacturing*, p. 198–207, Goteborg, Sweden; 1998.
6. OMG (Object Management Group). (2015). *OMG systems modelling language specification*. <http://www.omg.org/spec/SysML/1.4/>.
7. Ahsan Qamar, Torngren M., Wikander J. (2010) Integrating multi-domain models for the design and development of mechatronic systems, INCOSE 2010.
8. Zengshiting Zhang, (2016) Detecting and resolving Work-Space Congestions and Time-Space conflict through 4D - Modeling In the micro level, Master thesis 2016.