Doctoral thesis

Doctoral theses at NTNU, 2023:412

Shafi Khurieshi Mohammed

Model Based Definition for Robotic Assembly

NTNU
Norwegian University of Science and Technology
Thesis for the Degree of
Philosophiae Doctor
Faculty of Engineering
Department of Mechanical and Industrial
Engineering

NTNU
Norwegian University of
Science and Technology

Shafi Khurieshi Mohammed

Model Based Definition for Robotic Assembly

Thesis for the Degree of Philosophiae Doctor

Trondheim, December 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the Degree of Philosophiae Doctor

Faculty of Engineering
Department of Mechanical and Industrial Engineering

ISBN 978-82-326-7524-1 (printed ver.)
ISBN 978-82-326-7523-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

Doctoral theses at NTNU, 2023:412

Printed by NTNU Grafisk senter

Summary

This thesis deals with the product information requirements associated with automated robot programming for assembly operations. The first contribution of this thesis is the establishment of information flow from design to robotic assembly using STEP AP242Ed2 files. Model Based Definition (MBD) methodology is followed to capture the product manufacturing information (PMI) along with the product geometry during the design phase. MBD is the backbone of the Digital Thread (DT) in a connected enterprise. MBD is created using the second version of Application Protocol 242 of ISO 10303 standard released in 2020, also known as STEP AP242Ed2. PMI is added to the STEP AP242 files semantically by two methods. One way of adding the PMI is by using the standard entities defined in the STEP standard. Another method is using custom-defined Unicode strings in alignment with the industry-recommended practices. Using these two methods, almost all the PMI traditionally included in the 2D manufacturing drawings can be semantically added to the STEP AP242 files.

Another contribution of this work is extracting and reusing the product information, both geometric and PMI, for robotic task specification. The spatial relationships between the constituent parts in an assembly, GD&T, welding information, and surface finish annotations are semantically added to the STEP AP242 files. A motor assembly is used as a test case to illustrate the extraction of spatial relationships and their subsequent application in defining manipulator motion constraints for robotic assembly.

Welding symbols and GD&T are semantically embedded within the STEP AP242 files, facilitating the extraction of relevant PMI and associated part features. This extracted data, ranging from worst-case boundaries of mating features to welding annotations, plays a crucial role in robotic assembly operations, from estimating assembly forces to specifying welding constraints.

Furthermore, this thesis introduces a novel classification of handling features and criteria for gripper selection, leveraging product data from MBD files. The surface finish, threading, material, and other information extracted from the semantic annotations are instrumental in selecting grippers and planning grasping and handling tasks. The practicality of this approach is underscored through assembly experiments, where product data from MBD files is harnessed for robot programming.

This thesis contributes to the two significant areas of Industry 4.0, the establishment of DT and the adaptation of smart automation using robotic manipulators. Adopting the MBD methodology using STEP AP242Ed2 files facilitates the DT in the extended enterprise, including the suppliers and automatic constraint definition for robotic task specification with minimum human intervention.

Preface

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD) at the Norwegian University of Science and Technology (NTNU), Trondheim. This work has been performed at the Department of Mechanical and Industrial Engineering (MTP) under the supervision of Associate Professor Lars Tingelstad and Professor Olav Egeland.

Acknowledgments

First and foremost, I want to express my gratitude to Allah, the Lord of the Worlds, for all the blessings in my life.

I want to thank Associate Professor Lars Tingelstad for giving me the opportunity to do this Ph.D. I appreciate his valuable time and suggestions in guiding this thesis and keeping me motivated during the difficult period of the Covid pandemic. I am also thankful for his tips in taking care of kids as per Norwegian weather. I want to thank Professor Olav Egeland for giving me the chance to work in the Robotics and Automation Group and for his help in finalizing this thesis. I would like to thank my co-author Mathias Arbo for his productive collaboration, many interesting discussions, and his review comments.

I would also like to thank all my colleagues at Robotics and Automation Group for making the work days pleasant with interesting lunch and tea break conversations. I want to thank all the department administration members for their support and assistance. I appreciate the help from the library staff, especially Librarian Marianne Brødreskift, for her support and coordination in getting the various ISO and ASME standards. I would like to thank all my colleagues and friends from my previous company, especially my team manager.

I thank my parents for their support in every stage of my life and for inspiring me to seek knowledge. I want to thank my wife Farha for her patience and for caring for our kids in my absence. I want to express my appreciation for my children Ukasha and Umme Kulsum for being the joy in my life and for the innocent yet deep philosophical questions they pose. I thank my sister and her husband for

taking care of my son in India while I was stuck in Norway due to the pandemic. I want to thank my brother for his moral support and suggestions.

I want to thank all my friends, especially Abdullah, Waseem, and Sohail, for the tasty biryani, the Saturday evenings we spent making that biryani, and the fishing trips. I want to thank Mastan for his friendship and brotherhood. I cannot forget his help while moving to Norway from India for this Ph.D. position. I wish everyone has a friend like him. Finally, I would like to thank all the people who contributed directly or indirectly to the completion of this thesis, particularly all those researchers who contributed to this field before me. I hope I am adding something worth a grain of sand to its advancement.

بیگانه آمدم و بیگانه روم
خبر از خود ندارم که کیستم وچه کاره ام

Contents

List of Figures

List of Tables

List of Abbreviations

AM Application Module.

AP Application Protocol.

API Application Programming Interface.

ARM Application Reference Model.

ASME American Society of Mechanical Engineers.

BO Model Business Object Model.

BOM Bill of Materials.

CAD Computer Aided Design.

CM Core Model.

CNC Computer Numerical Control.

CTC Core Technical Capabilities.

DM Domain Model.

DO Domain Objects.

DT Digital Thread.

GD&T Geometric Dimensioning and Tolerancing.

HDF5 Hierarchical Data Format Version 5.

IAR Integrated Application Resources.

IGR Integrated Generic Resources.

ISO International Standards Organisation.

MBD Model Based Definition.

MBE Model Based Enterprise.

MBM Model Based Manufacturing.

MBSE Model Based Systems Engineering.

MIM Module Interpreted Model.

OEM Original Equipment Manufacturer.

OMG Object Management Group.

PbD Programming by Demonstration.

PMI Product Manufacturing Information.

PUMA Programmable Universal Machine for Assembly.

RPL Robot Programming Language.

SCARA Selective Compliance Assembly Robot Arm.

SDAI Standard Data Access Interface.

SE Systems Engineering.

SME Small and Medium Enterprise.

STEP STandard for Exchange of Product Data.

SysML Systems Modeling Language.

TDP Technical Data Package.

UML Unified Modeling Language.

UUID Universal Unique Identifiers.

XMI XML Metadata Interchange.

XML Extensible Markup Language.

Chapter 1.

Introduction

Digitalization and automation are two of the key technology enablers of Industry 4.0 in realizing its goal of intelligent robotic manufacturing in a connected digital enterprise [39]. As the industry is looking beyond Industry 4.0 [30], its adoption in Small and Medium Enterprises (SMEs) is limited. The European Union defines SMEs as enterprises with less than 250 personnel and turnover below 50 million Euros [31]. SMEs face unique challenges due to their limited finances and the availability of skilled personnel. Achieving the full potential of Industry 4.0 depends not only on its implementation in large enterprises but also on the implementation in SMEs, as they are part of the global supply chain and a large part of the economy [27].

The success of Industry 4.0 depends on how efficiently the design and manufacturing operations are integrated. Digitalization makes product data available for all stakeholders depending on their needs and requirements. The availability and completeness of product data enhance the efficiency of product manufacturing operations. A large amount of product information is created in the design phase of the product life cycle. The availability and reuse of this data at downstream operations are achieved by establishing a connected Digital Thread (DT) throughout the extended enterprise where an unbroken information chain makes the product data available for all the stakeholders depending on their needs.

Another aspect of Industry 4.0 is the increased automation of manufacturing operations. Robots are an essential part of future connected factories working in collaboration with other machines and humans. The robotic automation is suited for intermediate production volumes with medium variation in product variety, as shown in Fig. 1.1. Robotic automation matches the operating scenario of SMEs as they deal with higher product variety and lesser production volumes compared to large industries [101]. The robot must be programmed to perform each operation separately and for each component handled by that manufacturing unit. Due to

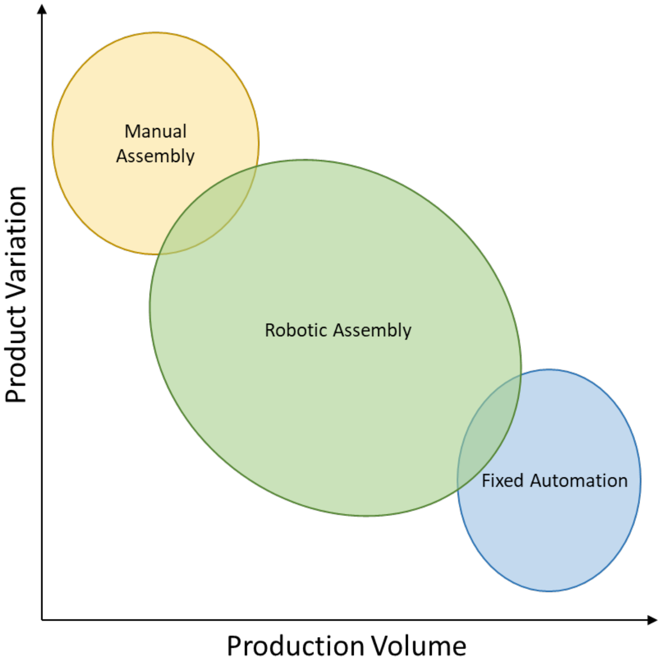


Figure 1.1.: Levels of automation with respect to production volumes and product variety.

this, the success of robotic automation is limited to large-scale industries, where the product variation is less and production volumes are higher.

1.1. Product Information - From Design to Manufacturing Automation

There is a gap between the design and manufacturing phases of the product life cycle. Mostly the design and manufacturing happen in silos, and the data created during the design phase is not readily available for the manufacturing phase. Even before the adaptation of Model Based Definition (MBD), most manufacturers are using native/neutral 3D CAD models along with the 2D manufacturing drawings [106]. There are significant gaps in product data exchange that break the DT and stop the realization of Industry 4.0 [35]. Due to these gaps, manual intervention is needed in the manufacturing phase, even in the case of automation. Manual programming is used to develop code to control the robotic manipulators, as shown in Fig. 1.2. The robot program has to be changed whenever the product design changes, which means more time and cost. Hence, this arrangement is

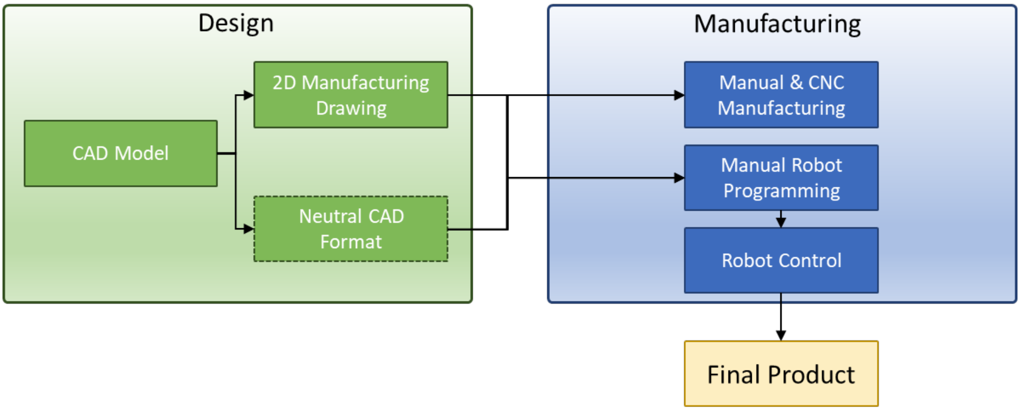


Figure 1.2.: Present industrial scenario with manual programming.

more suitable for large-scale manufacturing with limited product variation, like automotive manufacturing. For SMEs who deal with small quantity manufacturing of various products, this drastically increases the cost of automation. The adoption of robotic manufacturing by SMEs can be increased if the programming cost, both time and money, can be reduced.

This thesis aims to reduce this gap by avoiding or significantly reducing the need for manual robot programming. The ideal scenario is shown in Fig. 1.3, which fulfills the requirements of connected smart robots executing the tasks with low to minimal human intervention.

1.2. Robotic Assembly

Assembly is the culmination of all the upstream design and unit manufacturing processes into the final product [22]. It is one of the essential steps in production as it takes more than 50 % of total production time and 20 % of production cost [108]. Depending on the production volume, a product can be assembled manually, using fixed automation, or using robots. If the production volumes are low, assembly is done manually. Fixed automation using special-purpose machines is employed in the case of high volume production (a few million per year), like in the case of assembling rearview mirror buttons on automobiles. Flexible automation using programmable robots is used when both manual and fixed automation are not suitable options [130]. Even though some of the early industrial manipulators like PUMA (Programmable Universal Machine for Assembly) [36] and SCARA (Selective Compliance Assembly Robot Arm) [75] are designed for assembly operations,

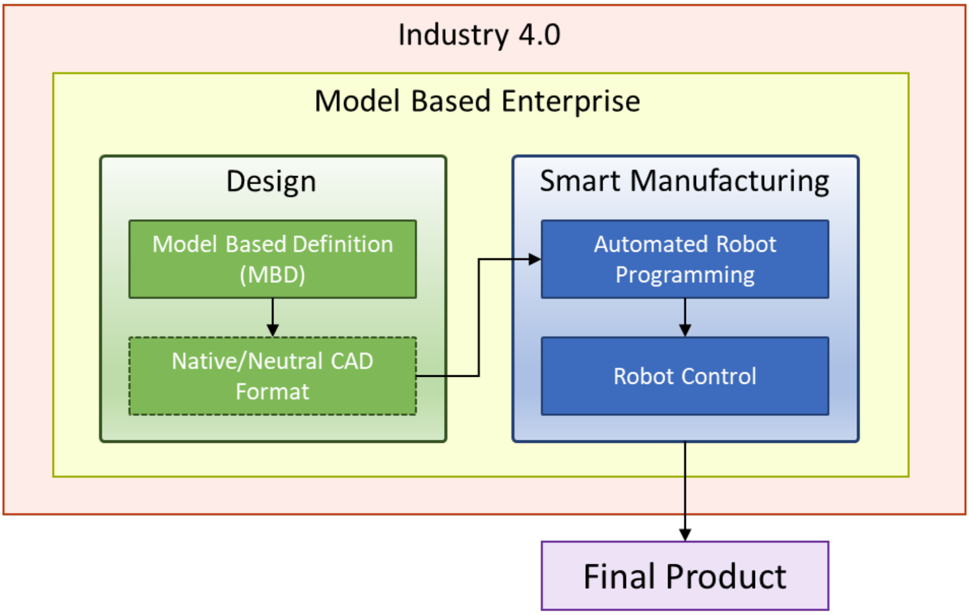


Figure 1.3.: Ideal process as per Industry 4.0 - with automated programming.

the use of robotic assembly in the industry is low compared to other operations like spot welding. Programming of robots for assembly operations is much more complicated than for spot welding [71].

1.2.1. Assembly Tasks

Schmidt [85, 74] classified the assembly tasks into five groups. These can also be regarded as the stage changes which a component undergoes during the assembly process [74]. The factors that affect these tasks are also identified by Schmidt [85]. The five types of assembly tasks are

- Store: The task associated with placing the components in the work cell and their presentation to the assembly robot before it handles the components. The main parameters that define this state are the initial position and orientation of the parts in the work cell with respect to a reference (world) coordinate frame. The components can be maintained in this initial position and orientation by simply placing on in the work cell or using fixtures. Store can also refer to the initial stage of bin-picking, where the objects are presented in the workcell in random poses to be handled by the manipulator.

- Move: The change in position and orientation of the component from the

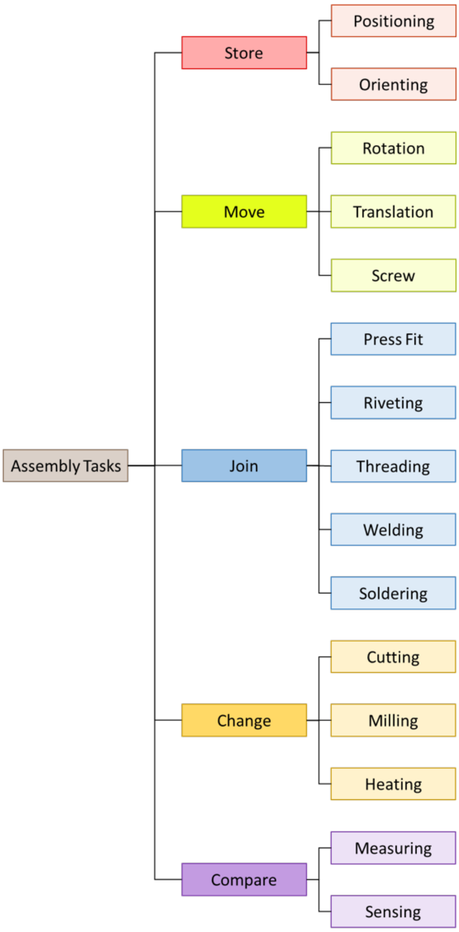


Figure 1.4.: Groups of assembly tasks [85].

initial store stage to the final assembly. The movement can be a rotation, translation, or screw motion. The input for this task is the initial and final position and orientation of a component. The movement depends on the mass, volume, geometry of the mating features, material, surface finish, and the manipulator kinematics. The mating features, material, and surface finish determine the contact condition, friction between the parts and the forces required to complete the task.

- Join: This group refers to the tasks associated with connecting or joining the mated parts. The connections can be temporary or permanent. The temporary connections include press fits, key joints, and threaded connections. The types of permanent joints are riveted joints, welding, and soldering.

Many times, the joining task is associated with moving, like in forming a screw connection between two components [74]. Successful completion of the task depends on the geometric shape, material, mass, volume, position, orientation, and process parameters like welding.

- Change: This group covers the tasks needed to change the shape or properties of a part in the assembly. The shape can be changed by cutting, chipping, milling, or punching. The components can be heated or cooled before the actual joining.
- Compare: The measurement and sensing tasks needed to detect the components, their stages, and other parameters during the assembly operation are grouped under this category. They include contact and non-contact tasks. The contact type includes probes that establish a contact between the component and the sensor/end-effector to measure a parameter. Non-contact type includes a visual sensor to measure the parameter.

Fig. 1.4 summarises these five task groups. This thesis considers tasks under the store, move, and join stages and their information needs.

1.3. Research Problem

This thesis answers the following research problem.

> How can the product data from the design phase of the product life cycle be used to enable efficient integration of design and automation in robotic assembly operations, thereby minimizing manual robot programming and facilitating seamless information exchange?

This research problem encapsulates the challenges and objectives outlined earlier, focusing on bridging the gap between design and manufacturing, leveraging digitalization, and enhancing the automation of assembly operations using industrial robots. By addressing this research problem, this thesis seeks to contribute to the broader objective of enabling connected and intelligent manufacturing systems and facilitating the adoption of Industry 4.0 practices across both large enterprises and SMEs.

1.3.1. Hypothesis and Research Questions

The primary hypothesis of this thesis is that product data, encapsulated within MBD files, can be reused for robot programming of automated assembly operations, thus establishing a connected digital thread spanning from design to robotic assembly and reducing/eliminating the need for manual intervention in data extraction and robot programming.

The research questions (RQ) that steer this investigation are:

RQ1: What specific categories of product data generated during the design phase are essential for the programming robotic assembly tasks?

RQ2: Which MBD file format most suitably encapsulates the required product information for robotic assembly?

RQ3: What methods can be employed to capture product data in the form of MBD files, and how to address the potential challenges that might arise for seamless data exchange?

RQ4: How can the product information from the MBD files be extracted and interpreted for robotic assembly operations?

RQ5: Can the product data extracted from the MBD files be used to establish a connected digital thread between design and automation by using it to program robotic assembly tasks?

By addressing these research questions, this study aims to contribute to the development of methodologies and techniques that bridge the gap between design and manufacturing, minimizing the manual effort in programming assembly robots, especially within the realm of SMEs. The overarching objective is to facilitate the integration of Industry 4.0 technologies within interconnected intelligent manufacturing facilities across a broad spectrum of enterprises.

1.4. Contributions of the Thesis

The current research harnesses the product data created during the product development phase for automated assembly operations using industrial robots. The following research objectives were met:

- Described a methodology to use product information from CAD files for robot programming and real-time robot control for assembly operations.
- Created semantic Model Based Definition (MBD) to include the information necessary for robotic applications in the STEP AP242 files.
- Described methods of extracting the relevant data from STEP AP242 files for constraint identification and task specification for robot programming
- Established an unbroken Digital Thread (DT) for product data from the design phase to manufacturing automation using STEP AP242 files

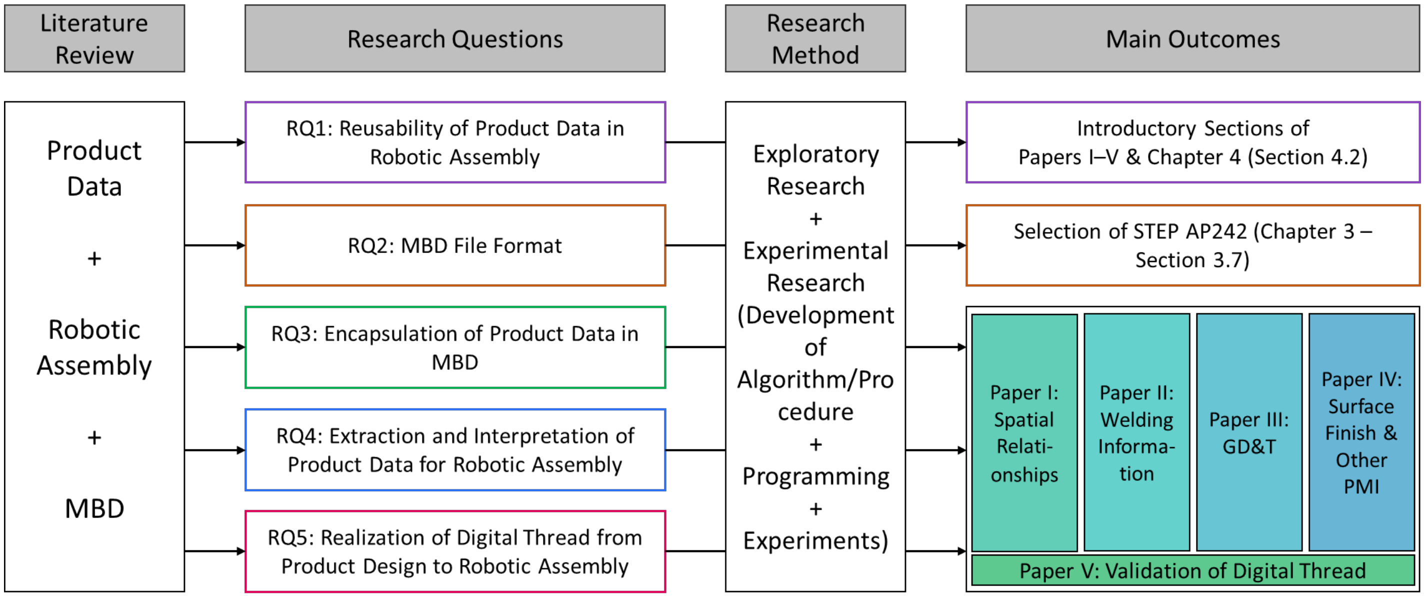

Literature Review
Research Questions
Research Method
Main Outcomes
Product Data + Robotic Assembly + MBD
RQ1: Reusability of Product Data in Robotic Assembly
RQ2: MBD File Format
RQ3: Encapsulation of Product Data in MBD
RQ4: Extraction and Interpretation of Product Data for Robotic Assembly
RQ5: Realization of Digital Thread from Product Design to Robotic Assembly
Exploratory Research + Experimental Research (Development of Algorithm/Procedure + Programming + Experiments)
Introductory Sections of Papers I–V & Chapter 4 (Section 4.2)
Selection of STEP AP242 (Chapter 3 – Section 3.7)
Paper I: Spatial Relationships
Paper II: Welding Information
Paper III: GD&T
Paper IV: Surface Finish & Other PMI
Paper V: Validation of Digital Thread

Figure 1.5.: Research Design and the Outcomes

1.5. Publications

This thesis is presented as a collection of published and submitted papers that were produced during the doctoral research. The research papers are listed below and attached in Appendix A. The research design in Fig. 1.5 shows the relationship between literature review, research questions, research methods, and the outcomes in terms of published papers.

Journal Papers

- Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. (2021) "Leveraging Model Based Definition and STEP AP242 in Task Specification for Robotic Assembly," Procedia CIRP. vol. 97, pp. 92–97
- Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. "Using Semantic Geometric Dimensioning and Tolerancing (GD&T) Information from STEP AP242 Neutral Exchange Files for Robotic Applications," International Journal on Interactive Design and Manufacturing (IJIDeM), 2023, https://doi.org/10.1007/s12008-023-01242-7.
- Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad, "Constraint Definition for Gripper Selection and Grasp Planning for Robotic Assembly using Product Manufacturing Information from STEP AP242Ed2 Files," Machines. 2022; 10(12):1230. https://doi.org/10.3390/machines10121230

Conference Papers

- Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. (2021) "Constraint Identification from STEP AP242 files for Automated Robotic Welding," 2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), 2021, pp. 277-282.
- Shafi K. Mohammed, Lars Tingelstad. (2023) "Robotic Assembly Using Product Manufacturing Information from Model Based Definition," (Will be submitted).

Apart from these research articles, the author has presented his work at the following conferences/forums.

- Presented a poster at "Digitalisation for Smart Processes and Product Desing: European Aluminium Innovation Workshop", held at Norwegian University of Science and Technology (NTNU), Trondheim, 12–13 June 2019
- Shafi K. Mohammed, (2019) "Leveraging Model Based Definition and STEP AP242 in Task Specification for Robotic Assembly," Presentation in Forum for Automatisk Produksjon, Trondheim by Norwegian Society of Electric and Automatic Control (NFEA), 28–29 August 2019

- Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. (2020) “Constraint Identification from STEP AP242 files for Automated Robotic Welding,” Presentation in “Constraint-Based Robot Programming (COBAROP) Workshop” at IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 24–30 October 2020.

1.6. Thesis Outline

This thesis is structured as follows. Chapter 2 presents the information requirements of robot programming for assembly operations and identifies the gaps in information flow. The Model Based Definition methodology and how it helps in attaching the product information to the part geometry is described in Chapter 3. Chapter 4 gives a brief overview of the STEP AP242 standard and maps the product data generated during the design phase to the corresponding capabilities of STEP AP242 for reuse in robotic assembly operations. The work done for this thesis is presented as summaries of published papers along with the methodology in Chapter 5. The contributions of the thesis are described in Chapter 6, followed by a discussion on the present work in Chapter 7. The concluding statements and scope of future work are given in Chapter 8. The research papers are added in Appendix A. Appendix B provides a brief overview of the STEP standard.

Chapter 2.

Product Data for Robotic Assembly

This chapter identifies the traditional robot programming methods and their drawbacks and presents constraint-based programming methodologies. This literature review delves into the use of CAD for robotic automation, scrutinizing the current landscape of robot programming, the critical role of PMI in robotic manufacturing, and the potential research gaps that offer opportunities for further exploration and advancement.

2.1. Robot Programming

The successful adoption of robots depends on the ability to program them to match the needs of that particular application. Traditionally, the programming of robots is done by programming by demonstration (PbD), teach pendant programming, and offline programming methods. Another programming method, constraint-based programming, is being used recently for the automated programming of robotic tasks. These methods are briefly described below.

2.1.1. Programming by Demonstration

Programming by Demonstration (PbD) is one of the widely used and intuitive methods for robot programming [33, 34]. In this method, the task to be performed is demonstrated to the robot by moving the end–effector using manual control. The positions of the end–effector and the forces acting on it during the demonstration are recorded. The manipulator then replicates the same movements and forces to complete the task. PbD can be done by teleoperating the robot or using vision/voice-based interfaces or kinesthetic teaching. Many researchers use machine learning algorithms to extract the skills from the demonstrations. These skills can be represented at two levels [12, 132].

- Low-level: The learning is mainly concerned with robot trajectories and forces.
- High-level: This level aims at the completion of complex tasks. The task is modeled as a combination of pre-defined motions. These individual robotic motions are derived from low-level learning.

2.1.2. Teach Pendant Programming

This method can be considered a special case of PbD [33]. A teach pendant is a handheld device used to control the manipulator pose [71]. The end-effector positions are defined using a teach pendant in this method. The teach pendant is used to specify point-to-point movement or a trajectory. The trajectory of the manipulator end-effector and its joint poses needed to execute a task are specified using a teach pendant. These values are stored in the controller memory and can be re-run to perform the task. The teach pendants have buttons and display screens that enable entering logical programs for controlling the manipulators [71, 83]. Researchers are developing smartphone-based virtual teach pendants using Android [16] and augment reality [117].

2.1.3. Offline Programming

The PbD and teach pendant programming require access to the physical robot during the programming process. In other terms, these two can be combinedly referred to as 'online' programming methods [33]. As the name suggests, offline programming is done away from the robot. The programs are run on simulated environments before deploying them on the robot for calibration, testing, and actual task execution. This method offers greater control over robotic motions, and the robot can be programmed to execute complex tasks. The robot can be programmed to make explicit movements or to execute tasks. Explicit robot programming languages (RPL) can be

- Controller-Specific Languages: The RPLs are developed for a specific controller used by the manipulator and are generally provided by the robot manufacturer
- General Purpose Languages: Generic computer languages like C can be used for robot programming. Commonly a library for robot-level functions is developed for this purpose.

The task-level programming is at a higher level than the explicit programming. The user can specify the task/sub-task directly rather than specifying the entire sequence of motions to complete a task. At this level, the manipulator must be able to decide the lower level subroutines like trajectories and grasp locations [71, 33].

The 3D models of the object, the manipulator, and the work cell form the basis of offline programming as they define the work environment and product for simulation.

These three traditional programming methods are not task-oriented as the manipulator motions are defined in either joint or work cell coordinates. This limits the suitability of these methods for automatic robot programming. The robot programs need to be changed with the product designs or work environment changes, even if the tasks to be completed are similar. Reprogramming of robots requires manual programming every time the product is changed, resulting in delays in deployment and increases in cost. These methods are more suited for automating simple repetitive tasks in large-volume production like spot welding in the automotive industry. Due to the limitations, these methods do not bring out the anticipated benefits of robotic automation for SMEs where the production volumes are small and product variety is relatively high.

2.1.4. Constraint Based Programming

The limitations of traditional programming methods can be overcome, and robot programming can be automated by adopting constraint-based programming. In this approach, the robotic task is specified in terms of relationships between the objects, the manipulator, and the environment. The manipulator can then be controlled to complete the task by satisfying the constraints. This method also enables task-level programming as the constraints are defined at the level of tasks/sub-tasks.

2.2. CAD for Robot Programming

It can be said that the developments in CAD and robotic assembly happened in parallel. In the 1970s, Popplestone et al. proposed a method of defining the relative positions of components of an assembly in the form of spatial relations [2, 97]. The relations between parts are specified using four mating types: against, coplanar, fit, and aligned. Mathematically, the spatial relationships are expressed as transformation matrices. Currently, most CAD software uses this or a similar method to capture the relative positions of components in an assembly. Based on this method, Ambler et al. developed a constraint-based task specification system for automated programming of robots for assembly operations called RAPT (Robot APT) in 1978 [1, 98]. APT (Automatically Programmed Tool) is a high-level programming language for numerically controlled machine tools developed by Doughlas Ross in 1956 [99]. RAPT uses spatial relationships to determine the positions of the parts by a constraint propagation method.

Prior to this, AUTOPASS (AUTOmated Parts ASsembly System) was developed by Lieberman and Wesley in 1977 [78]. This is one of the earliest high-level programming languages for robotic assembly. AUTOPASS uses the geometric relations between the parts in an assembly world model. The world model is updated at each step of the assembly operation. The user specifies the assembly sequence in English-like statements [78]. In both AUTOPASS and RAPT, the users provide the spatial/geometric relationships between the parts in an assembly manually.

Kaufman et al. [115] developed an assembly planning system, Archimedes 2, in 1996. Archimedes 2 uses 3D models from commercial CAD systems to plan the assembly sequences and robot programming. The system consists of two assembly planners and two assembly sequence animation facilities that use a robotic work cell model. The input to the system consists of Pro/ENGINEER (PTC Creo) models, which are then translated to ACIS readable files. In addition to the CAD models, the system needs a separate assembly file as input. This assembly file contains the list of part and sub-assembly filenames, transformation matrices specifying the position of parts in the work cell, information about part fits and joints, recommended assembly directions, tools needed, and in some cases, the part contacts. Based on the input assembly, the system generates textual plans, animations, and control code for robotic assembly [115, 119].

HighLAP (High Level Assembly Planning) system was developed by Wahl et al. in 2001. This system generates assembly plans from the CAD models, spatial relations, and the description of the robot work cell. Using an interface, the user specifies the symbolic spatial relations between the components, the grasp positions for each component, and the location of objects in the work cell. The generated assembly plans are decomposed into tasks and skill primitives for robotic assembly using the assembly-by-disassembly method. The tasks and skill primitives are expressed using UML [38, 125, 126].

Norberto Pires et al. [68, 67] introduced a CAD interface to program welding robots. This interface utilizes 3D CAD models of the workbench and the product to determine geometric information and the product's relative positioning within the work environment. Additional information, like the manipulator trajectories and the welding parameters, is added by a user. This collective data is stored in a DXF file and subsequently used to generate the robot program.

iTaSC (instantaneous Task Specification and Control) is a task specification system using the constraint-based approach developed by De Schutter et al. [73]. The task is specified as constraints between object and feature frames. This framework combines instantaneous task specification and estimation of geometric uncertainties using sensor data [100]. The end-effector (tool) and the object have a coordinate frame each. The features/surfaces of both the end-effector and the object that come in contact during the task completion are represented by another coordinate frame known as the feature frame. The task can be defined as a position constraint using transformation matrices from work cell coordinate frame to feature frame via end-effector and object frames. This method of defining the positional constraints is very similar to the method of describing the relative positions of components in an assembly [22].

eTaSL/eTC [29] is a development from the earlier work on iTaSC. eTaSL (expression graph-based Task Specification Language) is a task specification language based on Lua. It uses expression graphs, and tree-like data structures, to specify the geometric relations between the parts. eTaSL forms the constraint specification layer of the controller eTC (expression graph-based Task Controller). The other two layers of eTC are a solver layer that translates the constraints into a quadratic optimization problem and a numerical solver layer that solves the optimization problem [29].

Somani [111] and Perzylo et al. [5] of SMERobotics presented a constraint-based framework for robot programming using the 3D CAD models of the parts. Full CAD models are used for assembly, object detection, and pose estimation. The robotic tasks are specified in terms of constraint sets and solved as optimization problems using an exact solver. A shop floor worker specifies the constraints between components, similar to defining the mating constraints in CAD software [4]. Constraint nullspaces are formed from the mating constraints, and the system can plan assembly paths in the nullspaces using the solver[92, 111].

Arbo et al. [84] presented a system architecture that uses CAD information to infer assembly process parameters, particularly beneficial for tight-tolerance assembly situations. The required tasks from a task library are manually added to the CAD models, which were later converted into specific assembly skills (robot motions) along with the geometric primitives extracted from the CAD models. eTaSL/eTC is used for constraint-based task specification and a Mamdani-type fuzzy inference to decide on compliant assembly strategies. This enables the possibility of sensor-based motion and event triggering, which was further explored by Pane et al. [131, 94]. The relationship between mating parts is defined as 'CADconstraints' and is used to define the motion constraints for assembly tasks. These 'CADconstraints', along with the part dimensions and clearances, are used to extract sensor-feedback control strategies in force-based robotic assembly.

2.3. PMI in Robotic Manufacturing

Apart from geometric and topological information, the product design document captures other information required for manufacturing and assembly. The non-geometric information required to support product-related activities for the product's entire life cycle, like manufacturing, assembly, and inspection, is called Product Manufacturing Information (PMI). The most important types of PMI defined during the design phase [20] are given below.

- Dimensions: All the sizes, locations, and auxiliary dimensions are needed to define the product.
- Tolerancing: The allowable variations from the nominal dimensions are specified in the form of tolerances. Geometric tolerancing and dimensioning (GD&T) specifies the form, fit, and orientation tolerances.
- Welding Annotations: The welding process parameters are specified using the weld symbol.
- Surface Finish: The surface finish requirements are defined as surface finish symbols.
- Fasteners: The thread and other fasteners, like riveted joints, are specified under this PMI.
- Part List: A list of all the components in an assembly and their quantities are given as a Bill of Materials (BOM).
- Notes: Any additional product information, like manufacturing process, are added as separate text notes.
- Metadata: The information required to manage the product definition and manufacture are grouped under this category. The product information like part number, material, designer, and approver are some examples of metadata.

2.3.1. Relevance of GD&T

It is established in the literature that clearance and insertion length, as two pivotal parameters, determine the contact states during assembly operations and hence the force needed to complete the assembly [110]. These parameters also determine the conditions of jamming and the maximum tilt as elucidated by Whitney [129], and Haskiya et al. [37]. Part clearances also determine the contact and force models [81], as well as the necessary control methodology needed to complete the assembly operation [116, 15] successfully. The selection of the robotic manipulator and the success of the assembly operation also depends on the part clearances [118, 70].

In combination with manipulator accuracy and repeatability, part dimensions and tolerances impact the success of robotic assembly [28]. Part clearances and the manipulator uncertainty determine the control strategy for completing the assembly [19, 25, 118]. Proctor et al. [96, 95] suggested that using GD&T information along with manipulator uncertainty for motion planning will increase the accuracy of the robotic manipulators.

The part dimensions and GD&T information created during the design phase can be used to clearance between the mating features and can, in turn, be used in the robotic assembly in the following ways.

- estimate the contact states and jamming
- assembly forces needed
- suitability of a manipulator for the assembly task
- decide the control strategy

2.3.2. Relevance of Welding Annoations

Norberto Pires et al. [67] classified the welding parameters required for a good-quality weld into three groups based on the ease with which they can be controlled. They are as follows:

- Primary inputs: The variable parameters that can be adjusted during the welding, like welding speed, to achieve a good quality weld are considered primary inputs.
- Secondary inputs: The semi-fixed process-dependent parameters are classified as secondary inputs, for example, shield gas or filler material in the Gas Tungsten Arc Welding (GTAW) process.
- Fixed inputs: The parameters that depend on the product design, like geometry and material and fixed process parameters, are kept under this category.

As mentioned earlier, Norberto Pires et al. [68, 67] reused the weld geometry from CAD files for robotic welding. Larkin et al. [90] reused the information from neutral CAD files, like weld paths and other welding parameters, that were added to the 3D CAD models separately. Tran et al. [124] extracted the weld seam directly from native CAD files and combined it with the surrounding topology to plan collision-free manipulator paths for automatic robot programming.

Using an EXPRESS-based data model, Shen et al. [128] added the weld information like geometry, material, and quality requirements to STEP files. This information is reused for the programming of welding robots.

In manual welding processes, the welder intuitively adjusts for variations and errors to ensure the success of welding. It is crucial to consider these product variations in robotic welding processes to prevent issues related to product quality and rejections. Typically, sensors are employed in robotic welding to scan and extract the seam and groove geometry [67]. The planning of this sensing operation relies on human input.

2.3.3. Relevance of Other PMI

The gripping problem is one of the pivotal concerns in automated assembly, and the successful implementation of robotic assembly depends on it [88]. The properties of objects play a significant role in grasp planning and gripper selection, as the gripper interacts directly with the parts during handling and assembly. The literature comprehensively outlines the parameters that govern the selection and operational efficacy of grippers [93, 14, 21, 32].

Lotter [14] presented a comprehensive categorization of the object parameters influencing handling tasks into characteristics—like geometry, size, and physical properties—and behaviors—such as positional stability and transfer behavior, with some parameters like stackability being irrelevant to robotic gripping. Similarly, Fantoni et al. [32] identified critical parameters for gripper selection, drawing insights from the Design for Assembly principles. These parameters were classified into object parameters, encompassing both physical and geometric properties of the object, and operation parameters, which included factors related to feeding, handling, and placing of the object.

The object parameters are crucial in analytical grasp planning [40] and improve the success rate in the case of data-driven grasp planning methods [76, 112]. The essential object parameters that affect the grasp planning and gripper selection are [88, 79, 32]:

- Dimensional size
- Material: strength, hardness, density
- Weight
- Surface texture
- Special coatings
- Thread information [13]

- Assembly constraints and mating information
- Center of gravity
- Shape
- Part features: holes, surfaces

It is evident from the literature that part geometry and assembly constraints are being used for grasp planning from the CAD files [13, 107, 86, 102, 111].

2.4. Summary and Research Gaps

The above references capture the evolution of CAD-based robotic assembly systems, starting from the 1970s with the work of Popplestone et al. [2, 97] and leading up to recent developments by Pane et al. [131, 94] and Tran et al. [124]. The importance of PMI in robotic manufacturing is also established in the literature. Researchers are exploring ways to effectively utilize this information to improve the efficiency and accuracy of robotic manufacturing. Overall, the direction of research in this field is towards creating more flexible and intelligent robotic manufacturing systems that can handle various products and adapt to changes in the manufacturing environment. Despite the significant advancements in this field, the following research gaps are apparent from the above review:

- Automated Extraction of Assembly Constraints: All most all the systems require users to manually provide spatial/geometric relationships between parts in an assembly. There is a lack of research on the automatic extraction of the mating constraints and defining the relative positions of constituent parts in an assembly directly from the CAD models.
- Integration of PMI in Robotic Assembly: Proctor et al. [96, 95] proposed using GD&T from STEP files with the manipulator uncertainties for improving robotic assembly. The recent work by Pane et al. [131, 94] has made strides in using product information like clearances in combination with geometric information to infer assembly process parameters and decide on compliant assembly strategies. They have also hinted at using material properties for the force-based assembly. Apart from this, there is no mention of reusing PMI like GD&T, surface finish, and welding information directly from the design files.

The identified research gaps, the need for automated extraction of assembly constraints, and the integration of PMI in robotic assembly resonate with the inquiries posed in the research questions. Specifically, **RQ1** seeks to discern the essential product data categories from the design phase, which aligns with the gap concerning the manual provision of spatial relationships in assemblies. **RQ4**, which delves into the extraction and interpretation of product information from MBD files, directly addresses the highlighted need for automated extraction. Furthermore, the implementation aspects of reusing the product data for automation using MBD files are intrinsically tied to **RQ2** and **RQ3**, which explore the suitability of MBD file formats and their applicability as carriers of product data for automation. Bridging these gaps, as the literature suggests, could pave the way for a seamless transition from design to robotic assembly, establishing a digital thread (**RQ5**) and reinforcing the importance and relevance of the research questions formulated in this study. The details of MBD and its broader implications will be explored in detail in the subsequent chapter.

Chapter 3.

Model Based Definition

This chapter describes the Model Based Definition (MBD) methodology that will enable the inclusion of PMI in the CAD models. As evident from the previous chapter, MBD carrying both the product geometry and PMI will be an essential source of object information required to complete robotic assembly successfully. Section 3.1 describes the traditional practice of preparing 2D manufacturing drawings and the need for MBD. Sections 3.2 introduces the concept of MBD and presents the relevant definitions as per the applicable industry standards. In Section 3.3, different levels and types of MBD are described. The amount of PMI included in the MBD is given in Section 3.4. Section 3.5 discusses the structure and the arrangement of PMI in an MBD. Section 3.6 discusses the aspects of creating the MBD of an assembly and the inclusion of PMI needed for robotic assembly. Section 3.7 mentions the various file formats available for creating MBD and describes the criteria used in this thesis for selecting a suitable one.

3.1. 2D Manufacturing Drawings [1]

Traditionally, designers used to prepare the manufacturing drawings on paper by hand. These drawings were the input for manufacturing and inspection processes. With the advent of CAD systems, the designers started preparing the 3D models of the product using the CAD software. The designers then prepared the 2D manufacturing drawings from these 3D models by adding the PMI to the perspective views. Even though the CAD systems facilitated the geometric definition of the product, 2D drawings remained the source of product geometry and PMI for downstream manufacturing operations. There was no significant impact of CAD systems on the downstream processes. They continued to use the 2D manufactur-

[1] Many insights in this section stem from the candidate's experiences as a designer and engineering consultant.

ing drawings irrespective of whether those drawings were made by hand or from 3D CAD models. There are two significant limitations of using 2D drawings in terms of understanding and extraction of information.

As humans, our fundamental understanding of all the things in the universe is three-dimensional. We see the world in 3D, think in 3D, and the objects we use are 3D. The designers envision the product in 3D, and with the help of CAD, they define it in 3D. The manufacturer understands the product in 3D, and the final product produced is a 3D object. But the communication of the product definition from the designer to the manufacturer is 2D. This communication of product design using 2D manufacturing drawings is time-consuming as the designer has to prepare them from 3D models. The manufacturer has to spend time understanding the product from the perspective projections of the product. This process will also result in many miscommunications and further requires time for clarifications between the designer and the manufacturer. The miscommunications between design and manufacturing may lead to defective parts and product rejections if not addressed appropriately.

Another concern with 2D drawings is the extraction and recreation of PMI at the downstream operations. The manufacturer or the person responsible for the task reads the 2D drawing to extract the relevant PMI for his task. The relevant PMI for the specific downstream operations is identified and recreated manually after reading the entire manufacturing drawing. For example, the quality engineer must read the 2D manufacturing drawing, identify the critical quality parameters, and prepare the control plans. This manual recreation of product information at each stage of the downstream process is time-consuming and error-prone. Adopting the MBD methodology can eliminate these limitations of 2D manufacturing drawings or at least reduce them significantly.

3.2. Model Based Definition

The basic idea of Model Based Definition (MBD) is to make the 3D model the single source of product information by adding all the PMI to the product geometry. Though the 3D CAD models have been in use as a source of product geometric data for CNC machining for some time, the PMI is still manually interpreted and recreated from 2D drawings during manufacturing and other downstream operations. MBD enhances the reuse of product data created during the design phase and forms the backbone of the connected digital enterprise, also known as Model Based Enterprise (MBE). Therefore, MBD is the source of product geometric and manufacturing information that facilitates the establishment of the Digital Thread (DT) in the extended enterprise.

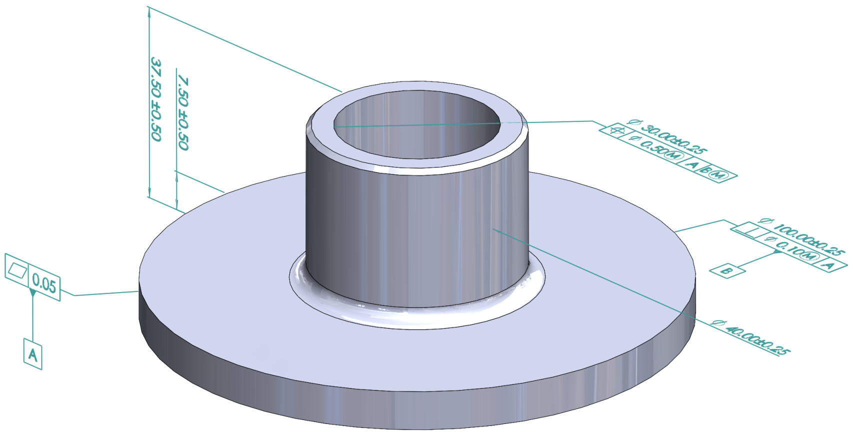


Figure 3.1.: MBD of a Part with GD&T Annotations [104].

MBD is the product documentation practice in which all the PMI along with notes and metadata are added to the 3D CAD model directly, thus making the 3D model the master of product definition [69]. The concept of MBD is first presented in the MIL-STD-31000A standard as part of the Technical Data Package (TDP) to facilitate the procurement of products by the USA Department of Defence [23]. As per the latest version, MIL-STD-31000B, the MBD contains all the data needed to define the product completely [24]. The ASME Y14.41 standard presents the concept of MBD as *Product Definition Data*. The first release of the ASME Y14.41 standard mainly dealt with the method of adding the GD&T to the 3D model [9]. The latest edition of this standard included surface finish and weld symbols. The definitions of MBD and *Product Definition Data*, respectively, from the latest releases of MIL and ASME standards, are close and refer to each other. These definitions, along with others from these standards, are repeated below.

3.2.1. Definitions

- Technical Data Package (TDP): “The authoritative technical description of an item. This technical description supports the acquisition, production, inspection, engineering, and logistics support of the item. The description defines the required design configuration and/or performance requirements, and procedures required to ensure adequacy of item performance. It consists of applicable technical data such as models, engineering design data, associated lists, specifications, standards, performance requirements, quality assurance provisions, software documentation and packaging details" [24].

- Model Based Definition (MBD): "The practice of using 3D datasets containing the exact solid representation, associated 3D geometry, and 3D annotations of a product's dimensions, tolerances, materials, finishes, and other notes to specify a complete product definition. (See ASME Y14.41)" [24]

- Product Definition Data: "Denotes the totality of product definition elements required to completely define a product. Product definition data includes geometry, topology, relationships, tolerances, attributes, and features necessary to completely define a component part or an assembly of part for the purpose of design, analysis, manufacture, test, and inspection (ASME Y14.100)" [10]

- Annotated Model: "A combination of model, annotation, and attributes that describe a product" [10].

- Fully Annotated Model: "A 3D CAD dataset, in which all necessary features to fully define the item (i.e., full design disclosure to include dimensions, tolerances, materials, notes, surface finishes, etc.) are included in a readily viewable form. (See ASME Y14.41)" [24]

- Model Based Manufacturing (MBM): The process of manufacturing using the product information (geometric, PMI, and even metadata) from the MBD [24]. Similarly, when carried out using MBD, other activities like inspection can be called MBx, for example, Model Based Engineering or Model Based Inspection.

- Model Based Enterprise (MBE): An organization that establishes model based operations for the entire product life cycle activities is a Model Based Enterprise (MBE). MBD forms the basis of all the product-related activities in such an organization. The product data is made available to all the stakeholders as per their requirements [87]. MBE can also be known as a connected digital enterprise.

- Digital Thread (DT): The IT infrastructure and the software tools needed to establish a connected digital enterprise along with the models are known as 'Digital Thread.' DT makes the models available to the stakeholders based on their needs irrespective of the time and place [7, 123]. In other words, DT makes the realization of MBE a possibility.

3.3. Levels and Types of TDP

Generation of product information starts with the start of product development. As the product matures from the concept phase to full-scale production, the product information increases and becomes complete. This reality is reflected in the levels of TDP defined by MIL-STD-31000B [24].

3.3.1. Levels of TDP

- Conceptual Level: The TDP captures the information needed to define the product concepts at this level. The MBD at this level consists of the conceptual model with very few annotations like envelope dimensions, material, generic tolerances, and title block information.
- Developmental Level: This level of TDP consists of product information available at the developmental phase of the product life cycle and supports analysis, prototyping, or limited production. At this level, the models are more detailed and carry dimensional annotations and notes to support prototyping in addition to the PMI from the conceptual level.
- Production Level: The TDP is complete at this level and includes all the production information to support full-scale manufacturing, inspection, and packing. The MBD is fully defined at this level with all the PMI needed in the downstream operations.

3.3.2. Types of TDP

This classification is based on the format in which the product definition is captured and shared among the stakeholders. The MIL-STD-31000B describes two types of TDP, 2D and 3D [24]. As the name suggests, 2D TDP consists of 2D manufacturing drawings. The 3D TDP is model-based, and the 3D model is the master of product data. The 3D TDP can contain 2D drawings derived from the 3D model or 3Di pdf files. The model only 3D TDP only contains the annotated 3D models and does not include any derived 2D drawings. That is, model only 3D TDP is the MBD, and from now on, these two are used interchangeably in this chapter. Table. 3.1 shows the types of TDP, delivery options, and the master source of product information.

3.4. Product Data in MBD

As the MBD is the source of product data, it should contain all the information traditionally included in 2D manufacturing drawings. The PMI added to the

Table 3.1.: Types of TDP [23]

Types of TDP	Deliverable	Master of Product Data
3D TDP	Model Only (MBD)	Annotated 3D Model
	Model, 2D Drawing/3Di pdf	Annotated 3D Model
2D TDP	Model and 2D Drawing	2D Drawing
	2D Drawing	2D Drawing

MBD is grouped as follows [24].

- Annotations: The PMI added directly to the part features is known as an annotation. These include dimensions, GD&T, surface finish, weld symbols, and material.
- Notes: The additional data related to the product and its manufacturing is added in the form of notes. The specific instructions for the manufacturing or handling of the product, organizational and industrial standards and procedures to be followed, and other information that cannot be added in the annotations are included in the notes.
- Metadata: The data about the MBD itself is the metadata. It facilitates the definition and administration of the TDP. The metadata contains part number, designer, revision, approver, approved date, and TDP number.

One aspect of preparing an MBD is the amount of product data to be included as annotations. Is adding all the information available in the 2D drawing to the 3D model necessary? Here is an example. As the 2D manufacturing drawing does not carry the exact geometric definition of the product, all the dimensions must be included in the drawing (whether nominal or toleranced). As the 3D model carries the exact product geometry, adding all the nominal dimensions is not beneficial as those dimensions can easily be extracted from the features in the model. Another crucial aspect in MBD preparation is deciding the amount of information to be shared with a specific operator or a stakeholder. The drilling operation does not find much use for weld symbols. So sharing the weld annotations with the drill machine operator does not make much sense. However, some aspects of MBD are needed by all the processes like the geometry and material. Both geometry and material will be useful for drilling and welding operations. The Minimum Information Model (MIM) addresses these concerns about the amount of data to be shared [91, 3]. It proposes identifying everyday information items needed by all the stakeholders and domain-specific information for each process. As shown in Fig. 3.2, by combining common elements with the domain-specific information, the MIM for a specific process can be created. Grouping and organizing the

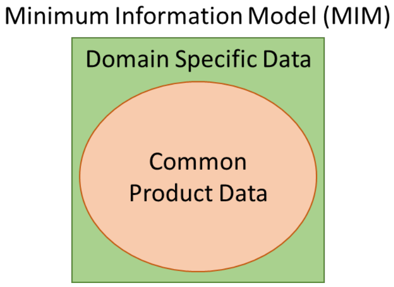


Figure 3.2.: Concept of Minimum Information Model (MIM).

product data will facilitate the efficient creation of MIM for each downstream process.

3.5. Organizing Product Information in MBD

The organization and presentation of product data were described in Appendix-B of the MIL-STD-31000A. Though the latest version of the standard does not contain this section, ASME developed it as a separate standard and released it as ASME Y14.41 in 2019. Depending on the way of consumption of MBD, the addition of PMI can be done in two ways.

- Presentation PMI: If the PMI is to be read and used by humans, then the PMI is referred to as presentation PMI. The latest ASME standards, Y14.41, and Y14.47, deal with the presentation of PMI for human users, i.e., visual aspects of the 3D model, organizing the various views of the model, and presenting the PMI in appropriate model views. Presentation PMI does not bring out the benefits of adopting MBD methodology as the PMI is still read and interpreted by humans.
- Semantic PMI: If the PMI is added to the features of the 3D model for identification, extraction, and interpretation by computer programs, then it is known as semantic PMI. This type of PMI will be much more useful and facilitate the automation in the product life cycle activities. Semantic PMI is supported by most of the commercial CAD software in their native format [82]. But the semantic PMI is converted to presentation PMI when the native CAD models are saved to STEP AP242 format. The main focus of this thesis is this type of PMI. Henceforth, unless otherwise specified, it always refers to semantic PMI whenever PMI is used.

Appendix-B of the MIL-STD-31000A presents a model schema to add annotations, notes, and metadata to the 3D model. The three significant aspects of the schema architecture are [23]

- Logical Groups: The PMI is divided into groups so that they can be made visible or hidden based on the product views. PMI elements can be grouped depending on the type, like GD&T or notes. The layers and groups tools in the CAD software can be used to form the logical groups of the PMI.
- View States: View states deal with the visual appearance of the 3D model like orientation, cross-section, display color, style, and exploded views for an assembly.
- Combination Views: The combination views are the final presentation of the MBD to the user. They are formed by the combination of the appropriate view states and logical groups of the PMI to arrange the MBD so that it will be easy to navigate and read by human users.

This MIL standard [23] also specifies naming conventions and orientation of the annotations and notes.

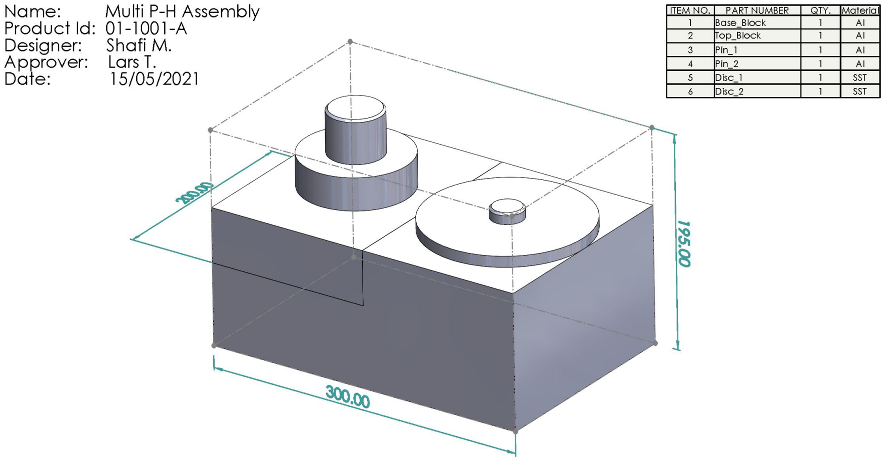
One of the best practices in preparing the MBD is forming the combination views and logical groups to match the downstream manufacturing processes like turning, milling, and drilling. Arrangement of the MBD as per downstream processes and in the order they happen will increase the re-usability of PMI in the organization.

3.6. Assembly MBD

Herron [69] discussed the process of creating the MBD of individual parts and assemblies as per MIL-STD-31000A and ASME Y14.41 standards in detail, along with the best practices. Most of the suggestions are very beneficial in the actual practice, even though they deal mainly with the visual presentation of the MBD. Some of those suggestions may not be well suited for automated information extraction for robotic assembly and welding operations. Fig. 3.3 shows an assembly MBD with some of the PMI and metadata.

3.6.1. Assembly Structure

- As this thesis is aimed at assembly operations, the MBD should be at the production level. Therefore, as per the standards, the geometry of all the constituent parts in an assembly must be fully defined.
- It is assumed that the components are added to the assembly model in the same order in which they will be assembled.



Name: Multi P-H Assembly
Product Id: 01-1001-A
Designer: Shafi M.
Approver: Lars T.
Date: 15/05/2021

ITEM NO.	PART NUMBER	QTY.	Material
1	Base_Block	1	Al
2	Top_Block	1	Al
3	Pin_1	1	Al
4	Pin_2	1	Al
5	Disc_1	1	SST
6	Disc_2	1	SST

Figure 3.3.: A view of an assembly MBD showing metadata, BOM, bounding box dimensions, and material details.

- The first constituent part added to the assembly will be a fixed part and will be positioned with respect to the assembly coordinate system.
- The subsequent constituent parts will be constrained with respect to previously added parts as per the intended function as opposed to the suggestion [69] of constraining with respect to the assembly coordinate system. Constraining the parts with respect to each other captures the design intent and enables the easy formation of tolerance chains between the parts and the identification of joints for kinematic analysis.

3.7. Selection of Appropriate File Format

Before moving forward, this is a suitable place to explain the logic behind selecting the STEP AP242 file format for implementing MBD for robotic assembly. The available 3D CAD formats for creating MBD can be grouped into native and neutral formats.

Native CAD Formats: Native CAD file formats are exclusive to the specific CAD software in which they were created. These formats encapsulate the maximum amount of design data possible, including geometry, topology, color, and layer information. They can also encompass metadata, assembly structure, PMI, and other data specific to the application.

The primary advantage of native formats is their ability to preserve all the features and attributes of the models as they were designed in the specific CAD software.

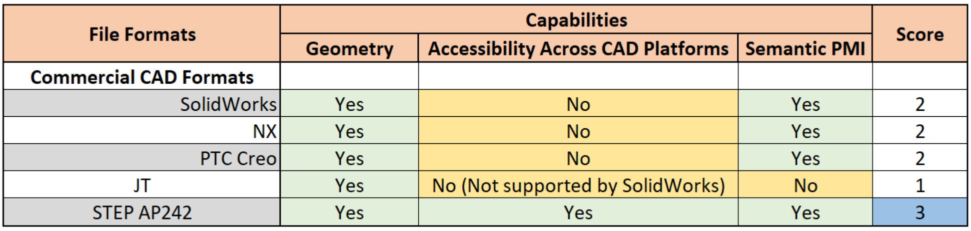
However, their limitation lies in their lack of interoperability, which can lead to difficulties in sharing files across different CAD systems.

Neutral CAD Formats: Neutral CAD file formats are designed to be used across different CAD systems. They are used to exchange data between CAD software, archiving data, or downstream applications such as manufacturing (CAM) or analysis (CAE). The main advantage of neutral formats is that they allow data exchange between different CAD systems, making them more versatile for collaborative and multi-software environments. The downside is that they may not support all the features of the native formats, and some information may be lost or approximated during the conversion process. The neutral CAD file formats suitable for implementing MBD are:

- STEP: This is a widely used format for exchanging 3D CAD data. It can represent both geometry and topology, as well as colors, layers, and assembly structure. STEP AP242 is a recent version that also supports the inclusion of Product and Manufacturing Information (PMI) [48].
- JT (Jupiter Tesselation): This lightweight and flexible 3D data format was first developed by Siemens and later adopted into an ISO standard, ISO 14306 [55]. It is primarily used for 3D visualization and can also facilitate collaboration and validation. It can represent both tessellated and exact geometry that supports different levels of detail for visualization, ranging from simple bounding boxes to detailed, shaded renderings [109].

A trade-off analysis is performed to select the best suitable CAD format for representing PMI. The CAD format should meet the following three requirements.

- The CAD format should have the required features to define the exact product geometry.
- It should be accessible across CAD/CAM platforms freely.
- The PMI should be represented semantically in the CAD file



File Formats	Capabilities			Score
	Geometry	Accessibility Across CAD Platforms	Semantic PMI	
Commercial CAD Formats				
SolidWorks	Yes	No	Yes	2
NX	Yes	No	Yes	2
PTC Creo	Yes	No	Yes	2
JT	Yes	No (Not supported by SolidWorks)	No	1
STEP AP242	Yes	Yes	Yes	3

Figure 3.4.: Trade-off analysis - score

These three requirements are equally important; hence, separate weights are not assigned to them for scoring. Fig. 3.4 shows the evaluation and scoring details. STEP AP242 file format has the highest score of 3 as it satisfies all three requirements. Based on this, STEP AP242 is selected to represent PMI for robotic manufacturing and assembly operations.

A detailed description of STEP AP242 is presented in the next chapter.

Chapter 4.

STEP AP242

This chapter describes the STEP neutral file format and explores its suitability for carrying product information from the design phase to robotic assembly. A description of STEP AP242 and the relevant aspects of the standard for assembly MBD are mentioned in Section 4.1. A brief discussion on assembly modeling using STEP AP242 is presented in Section 4.2 and maps the product data to the specific elements defined in the STEP AP242 standard. The limitations of this standard and the challenges faced during the implementation of MBD using STEP are mentioned in Section 4.3, along with the alternate options to overcome them. A brief description of the overall ISO 10303 (STEP) standard is offered in Appendix B.

4.1. AP242: Managed model-based 3D engineering

The MBDs prepared by the designers have to be shared with the downstream processes. They can be shared using the native CAD formats in which they were prepared. There are a few problems in sharing product data using native CAD formats. The downstream processes should use the same CAD software in which the designs were prepared. Moreover, the person performing the operations should be knowledgeable in using the CAD software. If the downstream process uses other software tools for its purpose, they may not be compatible with the native CAD formats. Hence, translation tools are required to convert from one format to another. Many SMEs serve various original equipment manufacturers (OEMs) as subcontractors. It will be an impossible financial hurdle for SMEs to purchase the licenses of all the CAD software used by different OEMs they serve. Hence, the need for a neutral exchange format that can facilitate product data exchange among all stakeholders. Such a neutral format should be capable of capturing both the product geometry and PMI available in the MBD. STEP AP242 format is specifically developed to serve this purpose.

The STandard for Exchange of Product data (STEP) is the informal name for the set of standards *ISO 10303: Automation systems and integration - Product data representation and exchange*. The *AP242: Managed model-based 3D engineering* is one of the latest application protocols that was released in 2014. The revised second edition was released in 2020, and the third edition will be published soon (awaiting approval[1]). AP242 grew out of the evolution of two earlier APs, *AP203: Configuration controlled 3D designs of mechanical parts and assemblies* and *AP214: Core data for automotive mechanical design processes*. AP203 was used by aerospace and defense, and the automotive industry used AP214. AP203 was the first of the two APs to be released in 1993, and it has the capabilities to capture 3D geometric, topological, and configuration management data of the mechanical parts and assemblies. AP214, first released in 2001, has the capabilities to define GD&T, kinematics, colors, and layers in addition to those of AP203. The latest editions of AP203 (2011) and AP214 (2010) had an overlapping scope, and further development does not differentiate them from each other. Hence, AP242 was developed, which replaced both AP203 and AP214 [77, 114, 47, 48].

4.1.1. Scope of AP242

AP242 defines the information models needed to represent the 3D design of mechanical products. This AP covers the geometry and PMI of both individual parts and assemblies, in other words, the definition of the MBD of the products. This AP replaced the earlier two APs, AP203 and AP214, and hence, covers the entire range of mechanical products, including automotive and aerospace sectors. This AP can capture product data throughout the product life cycle, including long-term archiving. The scope of AP242 includes product data management, mechanical design, 3D PMI, mating, kinematics, composite design, electrical harness assembly design, and additive manufacturing. AP242 is the most extensive and significant AP in the ISO 10303 standards.

4.1.2. Managed model based 3D engineering business object model

The business object model of STEP AP242 is given in Part-3001. The information required to define the product is specified in terms of capabilities and requirements in the form of Business Objects (BO). The BOs are divided into different logical groups. Some of the logical groups and some of the capabilities under them are

- Specification, Breakdown and Configuration:

[1] ISO\FDIS 10303-242 is in Stage: 50.00 - Final text received or FDIS registered for formal approval

  - Product Specification
  - Breakdown
  - Assembly Structure
  - Configuration
  - Part Occurrence
  - Transformation, Geometric Coordinate Space and Coordinate System
- Kinematics:
  - Kinematic Structure and Links
  - Kinematic Motion
- Characteristic:
  - Property
  - Material
  - Shape Association and Structure
- Document Management
- Rules and Requirements
- Process Plan
- Part Identification
- Geometric and External Element Reference
- Composites
- Activity and Work Management
- General Management Information

Assembly Structure Capability

As the name suggests, this capability allows the creation of the assembly structures by defining the relationships between either part views or between part views and part occurrences. This capability uses the following BOs.

- AssemblyDefinition
- AssemblyOccurrenceRelationship
- AssemblyOccurrenceRelationshipSubstitution

- AssemblyViewRelationship
- AssemblyViewRelationshipSubstitution
- CollectedPartRelationship
- CollectionDefinition
- GeometricalRelationship
- NextAssemblyOccurrenceUsage
- NextAssemblyViewUsage
- PromissoryAssemblyOccurrenceUsage
- PromissoryAssemblyViewUsage
- ReplacedPartViewRelationship

There are two types of assembly structures.

- Part occurrence based assembly structure: In this type of structure, the 'AssemblyDefinition' object represents the assembly, and the 'AssemblyOccurrenceRelationship' is used to relate the 'AssemblyDefinition' with the 'PartOccurrence' of the constituent parts.
- Part view-based assembly structure: This also uses the 'AssemblyDefinition' object for the assembly and the 'PartView' of the constituent parts is related to the 'AssemblyDefinition' using 'AssemblyViewRelationship' entitiy.

The relationship between the assembly and the constituent parts can be defined by specifying the cartesian transformation from the part coordinate system to the assembly coordinate system. Alternately, the relationship can be given by relating the geometric models. 'NextAssemblyOccurrenceUsage' entity is used to establish these relationships as shown in Fig. 4.1 [47].

Mating

The BOs under the mating capability are

- AssemblyJoint
- MakeFromRelationship
- MatedPartAssociation
- MatedPartRelationship
- MatingDefinition

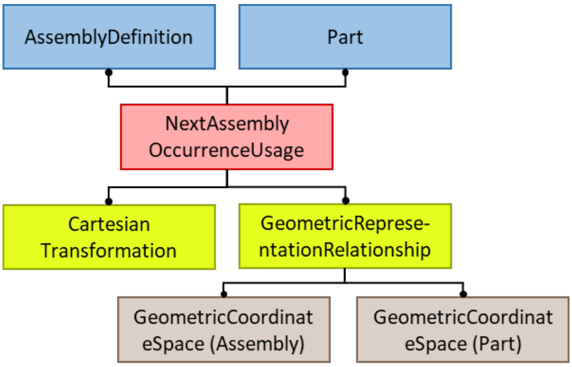

AssemblyDefinition
Part
NextAssembly OccurrenceUsage
Cartesian Transformation
GeometricReprese- ntationRelationship
GeometricCoordinat eSpace (Assembly)
GeometricCoordinat eSpace (Part)

Figure 4.1.: Definition of assembly structure.

- ProcessStateRelationship
- SameTimeMachiningRelationship
- ToolPartRelationship

These BOs capture information about the part manufacturing and assembly process like the mating type, joint type between the assembly components, tools for producing a part, and relating the raw material to the final product.

Transformation, Geometric Coordinate Space and Coordinate System

The BOs in this capability, like AxisPlacement, CartesianPoint, CartesianTransformation, Direction, GeometricCoordinateSpace, and RepresentationRelationship, enable the specification of geometric transformations between various coordinate systems in a part and an assembly.

4.1.3. Domain Model

The second edition of AP242 includes a Domain Model (DM) and is specified in Part-4442 in an effort to move towards a fully extended architecture. Ideally, the DM should map to the Core Model (CM). However, as AP242Ed2 does not have a CM, its DM refers to the ARM (specified in Part-442). The DM of AP242 defines the capabilities under 24 different domains like Activity, Breakdown, Composite Structural Shape and Structure, Document Management, Electrical Harness, Kinematics, Mating, Process Plan, Product Data Management, Shape Association and Structure, and Topology.

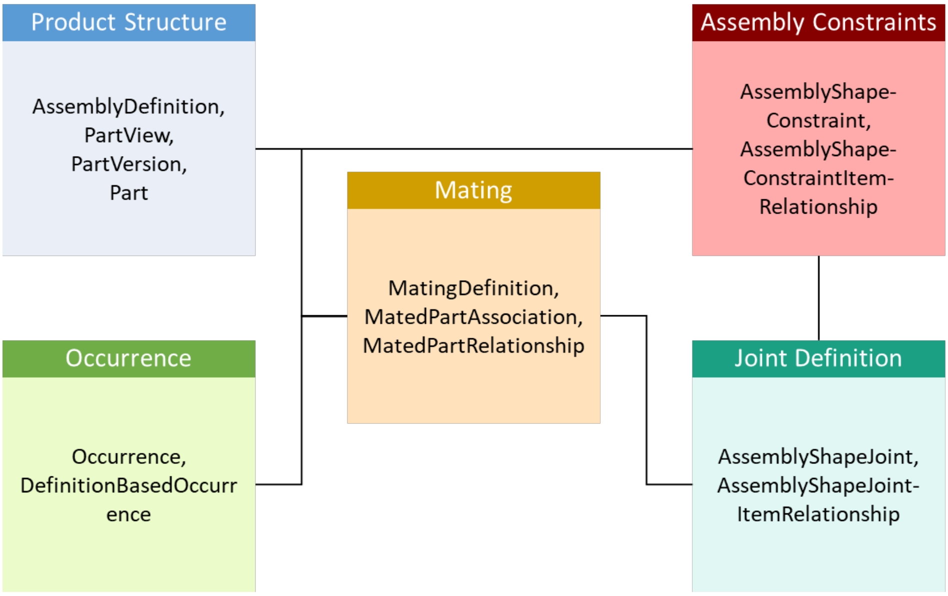


Figure 4.2.: Capabilities of Mating DM (Adapted from STEP standard [47]).

Mating Domain Model

Mating DM is an essential domain concerning the assembly process. It specifies the capabilities to make a detailed definition of the assembly joints, mating types, and constraints in an assembly. The domain objects are defined in a more detailed way than the BO specified in the BO Model. Fig. 4.2 shows the key capabilities of the Mating DM and the important entities used under each capability. These entities identify various assembly constraints, mating types, and joint types and enable the specification of mating and joint parameters.

4.2. Assembly Modeling Using AP242Ed2

The BO and DO from the BO Model and the Mating DM mentioned in the previous section are the fundamental entities that address the assembly structure and assembly process (mating and joining). These entities map to the ARM and finally to the Integrated Resources (both IARs and IGRs). Table 4.1 shows the Integrated Resources from the STEP standard that capture the product information.

Table 4.1.: STEP resources that enable the creation of semantic MBD for robotic assembly.

Product Information	Source in Manufacturing Drawing	Source in MBD	STEP AP242 Resources/Domain Models
Part Geometry	3D Model	3D Model	Geometry and topological representation (Part 42)
Part List (BOM)	2D Drawing/ PLM System	MBD/PLM System	Fundamentals of product description and support (Part 41)
Relative Positions	-	3D Model	Product structure configuration (Part 44)
Assembly constraints	-	3D Model	Kinematic and geometric constraints for assembly models (Part 109)
GD&T	2D Drawing	MBD	Shape variation tolerances (Part 47)
Welding Annotations	2D Drawing	MBD	Mating DM
Surface Finish	2D Drawing	MBD	Materials and other engineering properties (Part 45), Fundamentals of product description and support (Part 41)
Fasteners	2D Drawing	MBD	Mating DM
Notes	2D Drawing	MBD	Visual presentation (Part 46)
Metadata	PLM System	MBD/PLM System	Fundamentals of product description and support (Part 41)

4.3. Challenges of Using STEP AP242

The first challenge in the adoption of STEP is the role of commercial CAD software in implementing the STEP standard. All the CAD software has the capability to translate geometric definitions of the product from their native format to STEP. This aspect is mainly established from the earlier APs (AP203 and AP214), even though some information is lost during the translation to and from STEP files. The major limitation of the CAD software is in terms of PMI. All the CAD software that was tried converted the PMI to presentation PMI when translated to STEP AP242 but not semantic PMI. This limits the reuse of PMI at downstream operations using automated methods as presentation PMI needs human understanding and recreation of information as in the case of a 2D manufacturing drawing. The lack of this functionality limits the implementation of MBD and hinders achieving the goal of complete automation at the downstream operations.

Third-party software can be used to semantically translate all the PMI from native CAD format to STEP AP242. However, this will increase the financial burden on the organizations. Other options are developing in-house applications to perform the task or using JSDAI API to add the semantic PMI to the generated STEP files. These are not viable solutions as it moves the manual effort from the manufacturing stage to the design phase. It will be ideal if the CAD software provides the functionality to translate PMI semantically to STEP files as they already translate part geometry and presentation PMI.

The STEP AP242 standard has the capability to include the assembly constraints and GD&T information using the standard entities. Other information, like welding, can not be included using standard entities. Currently, the standard is not fully defined, and the third edition of the standard is under development. The second edition presented a Domain Model (DM) that includes mating capabilities. This domain model is supposed to be mapped to a Core Model (CM), which is not included in this edition of the standard. It is hoped that the third edition will include the core model and all the capabilities identified in the mating domain model will be mapped to the entities in the core model.

Chapter 5.

Summary of Papers

This chapter offers a consolidated presentation of the research undertaken, structured around the individual papers. While each paper offers a unique perspective and addresses specific facets of the research questions, an overarching methodology binds them together. Although implicit in the papers, this methodology is elucidated here to provide a cohesive understanding of the research. The paper summaries present a systematic approach where each paper, while focusing on distinct types of product data, consistently addresses research questions RQ1, RQ3, RQ4, and RQ5. This structured approach not only underscores the depth of the investigation but also highlights the interconnectedness of the various research components in the broader context of the thesis.

5.1. Methodology

This section presents the methodological approach employed to harness product data for robotic assembly operations, building upon the foundational methodology introduced in Paper I.

Fig. 5.1 illustrates the proposed methodology. Central to this research is the reuse of product information—geometry and PMI—originating from the design phase for the automated programming of robotic assembly operations. The methodology is rooted in constraint-based programming formalism, categorizing task constraints into three distinct groups, similar to the division by Somani [111]. The three types of constraints are described below.

- Object Constraints: These pertain to the assembly and its constituent parts. The process requirements for the successful completion of the assembly operation are also included in this category. Derived from product geometry and PMI, these constraints dictate parameters like position, velocity, and forces on the end-effector.

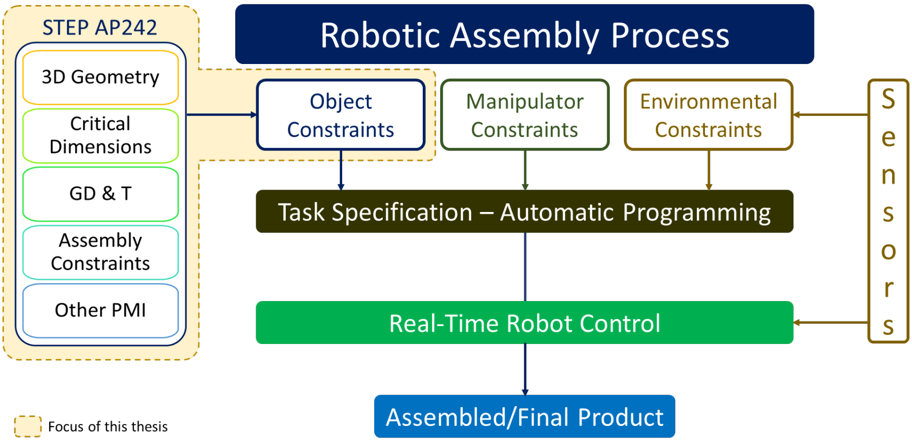


Figure 5.1.: Methodology of constraint based programming for robotic assembly [103].

- Manipulator Constraints: These encapsulate the kinematic and dynamic properties and limitations of the manipulator, such as joint position, velocity, and torque limits. The manipulator constraints are well-researched and standardized in the industry.
- Environmental Constraints: These constraints are related to the design of the work cell and the assembly line. This type of constraint includes the positions of various objects and fixtures and mainly determines the trajectory planning for task execution.

The primary focus of this research is the extraction of object constraints from product data using STEP AP242 files. Product information, embedded during the design phase using the Model Based Definition (MBD) methodology, is converted from native CAD to the STEP AP242 neutral format. This format ensures a machine-readable, platform-independent representation of product geometry and PMI. Extracting this data from STEP AP242 files facilitates the definition of object constraints. These object constraints, in combination with manipulator and environmental constraints, specify the assembly task.

However, the inherent limitations of STEP files in conveying PMI posed challenges. To circumvent these limitations, the following structured approach was adopted:

Adherence to International Standards and Practices: As this study started with mapping the essential product information for robotic assembly to the information captured in 2D manufacturing drawings, it was logical to adhere to the content and the manner in which it was being done traditionally. For this purpose, this study intends to meet the specifications given by relevant international standards from ISO, AWS, and ASME in terms of including the PMI in STEP AP242 files. Industry practices, such as those recommended by CAx, were also considered, with modifications to ensure comprehensive and semantic annotations.

PMI Integration in STEP AP242:

1. Each PMI type was scrutinized against current international standards, followed by a comparison with STEP AP242 capabilities. This comparison led to the identification of three scenarios:

    a) STEP AP242 does not have the entities defined for the required PMI, as the weld annotations

    b) STEP AP242 has the schema and entity definitions matching the PMI requirements, for example, GD&T

    c) STEP AP242 defines the capabilities for a particular PMI, but they do not meet the latest versions of the relevant standards, as in the case of surface finish symbols.

2. A systematic process of forming Unicode strings was developed for all the PMI in general and weld annotations in particular. Each PMI category was then mapped to a specific format or structure of Unicode string. This process ensured consistency and ease of interpretation.

3. In the subsequent two cases, where STEP AP242 offers partial or complete schema and entity definitions to carry the PMI, appropriate entities and methods of relating them in the STEP files were given, as in the case of surface finish PMI in Paper IV.

PMI Extraction:

1. For the Unicode string annotations, specialized parsers were developed. These parsers were designed to read the Unicode strings embedded in the STEP files, interpret them, and extract the encapsulated product data.

2. For the PMI added, using standard entities, pseudo-algorithms/processes were developed to extract the relevant product data for robotic assembly as described in Papers III and IV for GD&T and surface finish information.

Validation: The proposed methods were validated through coding, simulation, and experiments, as demonstrated in the use cases and experimental results.

In essence, this methodology facilitated the extraction of nearly all object constraints from STEP AP242 files for assembly tasks.

5.2. Summary of Paper I

This section is based on the first paper that deals with the assembly constraints from STEP assembly files and uses them for robot programming of assembly tasks. The identification of the spatial relations between the parts in an assembly forms the first step in our approach toward reusing PMI from MBD for robot programming. The relative positions of constituent parts in an assembly are included using standard STEP entities. The assembly constraints from STEP AP242 files were extracted and used to define the motion constraint on the robot during the assembly process. The entire process is demonstrated by simulating the assembly of a motor use case. This paper partially answers **RQ3**, **RQ4**, and **RQ5** regarding the addition, extraction, and reuse of assembly constraints for robotic assembly.

STEP AP242 [47] defines 9 standard entities for assembly constraints to capture the relative positions of constituent parts in an assembly. They are:

- *Fixed_constituent_assembly_constraint*: This is applied to the part which is fixed in the assembly and all other parts are placed with respect to this part. The coordinate system and origin of this part coincides with the world-coordinate system and the origin of the assembly.
- *Parallel_assembly_constraint*: This is used to define the parallel relationship between the mating features of the parts. This is applicable to lines and planes.
- *Parallel_assembly_constraint_with_dimension*: This is a subtype of parallel_assembly_constraint with a 'distance_value' to define the distance between the parallel features.
- Surface_distance_assembly_constraint_with_dimension: This is used to define the normal distance between lines and planes of the mating parts.
- *Angle_assembly_constraint_with_dimension*: This defines the angle between the line and plane features.
- *Perpendicular_assembly_constraint*: This is used to apply the perpendicularity constraint between the mating lines and planes.
- *Incidence_assembly_constraint*: This is used when a line is incident on a plane or to define the concentricity of circular, cylindrical, conical, or spherical features.
- *Coaxial_assembly_constraint*: This constraint is used when the axes of cylindrical or conical features have the same direction.

Table 5.1.: Assembly Constraints From STEP AP242 and Corresponding Mates from Commercial CAD Software

Assembly Constraint Entity	SolidWorks	NX
Fixed_constituent_assembly_constraint	Lock	Fix
Parallel_assembly_constraint	Parallel	Parallel
Parallel_assembly_constraint_with_dimension	Distance	Distance
Surface_distance_assembly_constraint_with_dimension	Distance	Distance
Angle_assembly_constraint_with_dimension	Angle	Angle
Perpendicular_assembly_constraint	Perpendicular	Perpendicular
Incidence_assembly_constraint	Coincident	Touch Align
Coaxial_assembly_constraint	Concentric	Concentric
Tangent_assembly_constraint	Tangent	Touch Align

- *Tangent_assembly_constraint*: This defines the tangency of circular features with lines, planes, or other circular features.

These assembly constraints correspond to the mates available in commercial CAD software as shown in Table 5.1. These entities are sufficient to completely constrain the parts in an assembly with respect to each other. However, even though the STEP AP242 standards support assembly constraints, when the assembly is exported as a STEP file from CAD software, e.g., SolidWorks or Siemens NX, the constraints are not explicitly available in the STEP file.

The constraint information, including parts in the assembly, the relative position of the parts to each other, assembly constraints, mating features of the parts, and their positions, is extracted from the STEP AP242 file and used for task specification. The steps in this process are represented here in the form of a flow chart in Fig. 5.2.

The motor assembly is used to show the possibility of reuse of assembly constraints for robot programming. The insertion of the rotor in the motor housing of a motor assembly, shown in Fig. 5.3, is used to demonstrate the generation of robot motion derived directly from STEP AP242 assembly constraints translated into eTaSL tasks.

5.2.1. Contributions of Paper I

This paper explored the integration of design with robotic assembly systems by harnessing the information encapsulated within MBD files. The process of extracting assembly constraints and reusing them for robot programming for assembly operations, combined with the practical application demonstrated through the motor assembly test case, has established STEP AP242 files as an invaluable

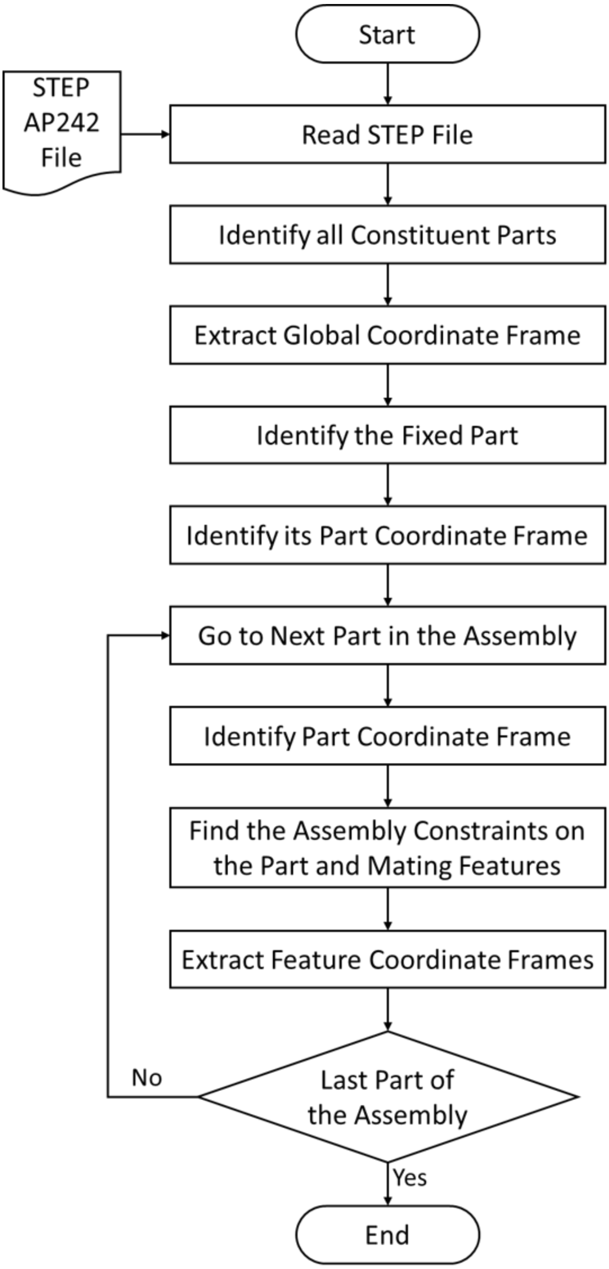

Start
STEP AP242 File
Read STEP File
Identify all Constituent Parts
Extract Global Coordinate Frame
Identify the Fixed Part
Identify its Part Coordinate Frame
Go to Next Part in the Assembly
Identify Part Coordinate Frame
Find the Assembly Constraints on the Part and Mating Features
Extract Feature Coordinate Frames
Last Part of the Assembly
No
Yes
End

Figure 5.2.: Process of Extracting the Constraint Information from STEP AP242 Files

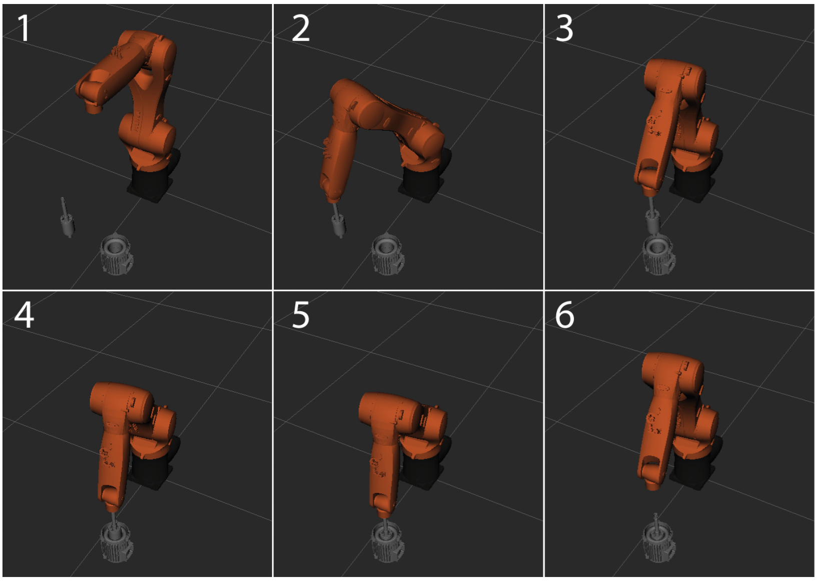


Figure 5.3.: Assembly sequence, frames 1 and 2 show the part being grasped and lifted above the housing, frames 3-6 show the motion being guided by the assembly constraints for inserting the rotor.

source of product information for automated robot programming.

Below are the contributions of this paper:

- Use of STEP AP242 Exchange Files: This paper establishes that STEP AP242 files can be used as a source of critical information for robotic assembly applications, as it provides not just the geometric details of the product but also the product structure and assembly constraints.
- Constraint Information: The constraint information, including parts in the assembly, the relative position of the parts to each other, and assembly constraints, is extracted from the STEP file and passed to the eTaSL framework for task specification.
- Assembly Constraints to Motion Constraints: This paper maps the assembly constraints to the type of contact between the mating geometric features, thus converting the mating constraints from STEP assembly files to the motion constraints for robotic tasks.

5.3. Summary of Paper II

After establishing the usability of assembly constraints from STEP AP242 files, the second paper delves into the potential of the STEP AP242 standard as a tool for extracting essential product data, specifically for the programming of welding robots. The **RQ3** and **RQ4** are answered in this paper with respect to welding annotations. This paper also provides answers to the **RQ5** partially by identifying the weld seam and other process parameters from the weld annotations.

This paper discusses the following three methods for incorporating welding information into the STEP assembly files.

- Using Custom EXPRESS Schema
- Using Unicode/Custom Code String
- Using Standard Entities from STEP AP242

A separate EXPRESS schema is defined by Shen et al. [127] to capture all the welding information like geometry, quality requirements, and even the information like the designer and the approving authority. The newly defined schema is exhaustive, encompassing all welding details, though some information is not pertinent to robot programming.

The other two methods work with the existing capabilities of the current standard. The paper presents a detailed description of forming a Unicode/Custom code string to include all the information that can be added to a traditional 2D manufacturing drawing as a weld symbol. The addition of welding information is specified by *AWS A.24 edition 2012 Standard Symbols for Welding, Brazing, and Nondestructive Examination* and *ISO 2553:2013 Welding and allied processes – Symbolic representation on drawings – Welded joints* standards.

A semantic Unicode/Custom code string is formed per the CAx recommendations of "PMI Unicode String Specification Examples and Mapping Strategies" [18]. The AWS and ISO standards use 39 symbols for welding annotations. Of these 39 symbols, 22 symbols are mapped to existing Unicode characters from various Unicode character sets, and custom codes are defined for the remaining 17 symbols. A method to form the custom codes and the annotation string is described. The weld symbol and its corresponding Unicode string are shown in Fig. 5.4.

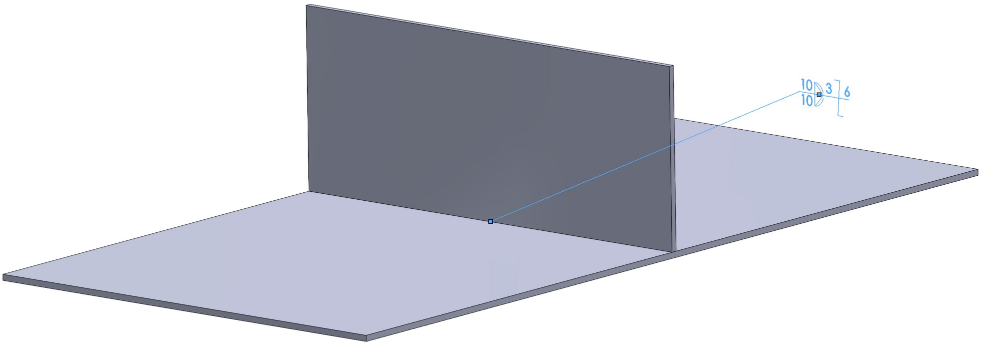
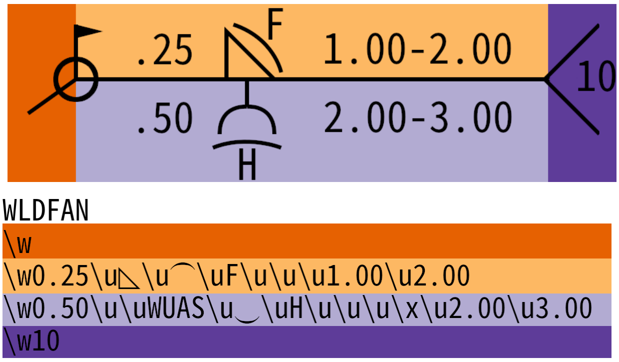


Figure 5.5.: T-joint with welding annotation.



```
WLDFAN
\w
\w0.25\u◺\u⏜\uF\u\u\u1.00\u2.00
\w0.50\u\uWUAS\u⏝\uH\u\u\u\x\u2.00\u3.00
\w10
```

Figure 5.4.: Regions of a welding symbol and the corresponding regions highlighted in the Unicode string. The Unicode string is split into multiple lines for clarity.

The last method proposes using standard entities available in the STEP standard. However, its major limitation is that the standard does not have the entities and schemas defined to capture all the welding information. It is hoped that the standards committee will expand the 'Mating Capabilities' domain from the second edition to enable the semantic inclusion of weld annotations.

Fig. 5.5 shows a weld symbol, and its equivalent Unicode string is added to the STEP file using a 'TEXT_LITERAL' entity. 'TEXT_LITERAL' entity as shown below.

```
#14=TEXT_LITERAL('Welding Annotation','WLDNNN\w\w10.0\u\u25FA\u23DC
\u\u\u\u3.0\u6.0\w10.0\u\u25F8\u23DC\u\u\u\x\u\u\w',$,$,$,$);
```

The seam geometry can be readily identified as the Unicode weld annotation is directly attached to the weld seam. This seam geometry, along with the spatial relationships derived from the assembly constraints (Paper I) and weld parameters extracted from the Unicode weld annotations, can completely define the welding task for robot programming.

The research has been implemented using the JSDAI API in Java. This Java-based API facilitates the reading and writing of STEP files or any EXPRESS-based data model. A parser is developed to interpret the Unicode string and extract the required weld parameters.

5.3.1. Contributions of Paper II

The methodologies presented here, from the innovative use of Unicode strings with the capabilities of the second edition of STEP AP242, offer a promising pathway to achieve integration of design data with robot programming for downstream manufacturing processes. The successful extraction of weld information using this method proves that MBD, implemented using STEP AP242 files, can be a source of complete product information, including the geometry and PMI required for programming welding robots. The main contributions of this paper are listed below.

- The welding annotations are added as per the standards followed by the welding industry, that is, *AWS A.24 edition 2012 Standard Symbols for Welding, Brazing, and Nondestructive Examination* and *ISO 2553:2013 Welding and allied processes – Symbolic representation on drawings – Welded joints.*
- All 39 symbols from the traditional weld annotations are mapped to the Unicode characters and custom codes to make the welding annotations fully semantic.
- The weld geometry is identified and extracted easily, as the weld annotation is directly attached to the part feature.
- The welding process parameters are extracted, which can be used to define the seam geometry, position, orientation, and motion constraints on the welding gun.

This paper also suggests using welding annotations in combination with GD&T to estimate the possible deviations in seam and weld parameters. The GD&T annotations were discussed in detail in the following paper (Paper III).

5.4. Summary of Paper III

This paper is focused on the reuse of GD&T information from the design phase for robotic automation of manufacturing operations. Before describing the ways of adding GD&T to STEP AP242 files, the paper presents the relevant literature that establishes the importance of GD&T for robotic assembly and welding operations (**RQ1**). The paper presents two primary methods for adding GD&T annotations to the STEP AP242 files. The first method uses Unicode strings, and the second utilizes STEP AP242 Ed2 standard entities. Both methods aim to semantically represent the GD&T information, making it reusable for robotic applications, thus partially answering **RQ3**.

The first method utilizes Unicode strings to add GD&T information, drawing inspiration from the CAx recommendations of *PMI Unicode String Specification Examples and Mapping Strategies* [18]. The addition of GD&T to the STEP AP242 files conforms with the international tolerancing standards like *ISO 1101* [53] and *ASME Y14.5: Dimensioning and Tolerancing standard* [11]. The possible combinations of individual dimensions with various tolerances, datum callouts, and Feature Control Frames (FCF) can be added to the product design are considered.

The method presented in this paper builds upon the one presented in Paper II. Unlike the CAx recommendation, the entire Unicode string is made semantic using this method. Fig. 5.6 shows a size dimension with equal bilateral tolerance and FCF for position tolerance mapped to the equivalent sections of the Unicode string. This approach is particularly beneficial due to its simplicity and directness, as the representation of GD&T in textual format makes it easily interpretable and accessible.

The second method employs entities defined explicitly in the STEP AP242 Ed2 standard as recommended by CAx in *"Representation and Presentation of Product Manufacturing Information (PMI) (AP242)"* [17]. The STEP AP242 Ed2 is analyzed to establish the applicability of the CAx recommendation as it was developed for the first edition of the STEP AP242 standard. It was found that no significant changes to the tolerancing entities in the second edition prevent the applicability of this CAx recommendation.

The article delineates two processes (**RQ4**) tailored to extract this information from Unicode strings and standard entities. Extracting relevant tolerance information from Unicode strings is straightforward using a parser. The parser can read and interpret the textual data, converting it into actionable insights for robotic applications. However, the extraction process demands a more nuanced approach for files that utilize the standard entities of STEP AP242 Ed2. It necessitates the identification of specific tolerance entities and their subsequent linkage to the geometry of part features. While more complex, this method ensures a compre-

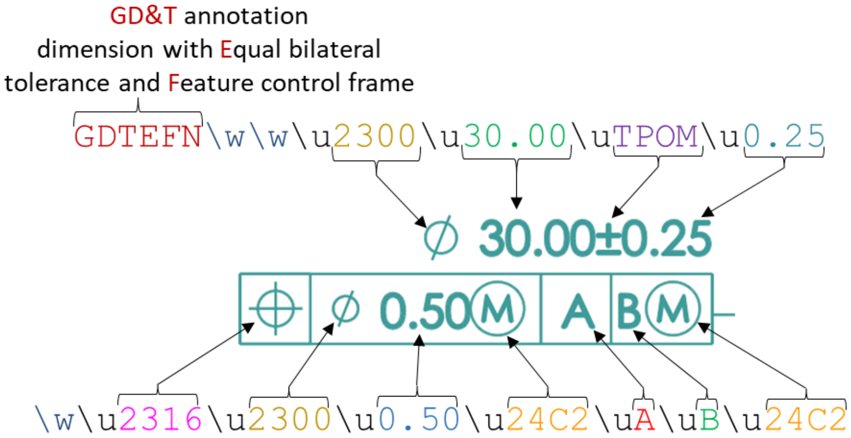

GD&T annotation
dimension with Equal bilateral
tolerance and Feature control frame
GDTEFN\w\w\u2300\u30.00\uTPOM\u0.25
⌀ 30.00±0.25
⌀ 0.50 Ⓜ A B Ⓜ
\w\u2316\u2300\u0.50\u24C2\uA\uB\u24C2

Figure 5.6.: GD&T Annotation with the corresponding Unicode string.

hensive extraction of GD&T information, capturing every intricate detail vital for manufacturing.

The process is applied to the Rotor and the Housing of the motor assembly (**RQ5**). The Rotor and Housing clearances are calculated using the dimensions and tolerances extracted from STEP AP242 files. The assembly process is split into various stages, starting from the initial approach to the final assembly stage, depending on the clearance ratio between the mating features of the components as shown in Fig. 5.7. For robot programs, changes in clearances between the mating features during the insertion process can necessitate a change in the control strategy.

The tolerances influencing welding are divided into two categories: the groove tolerances that determine the edge preparation by influencing root gap, bevel angles, and root face, and the generic tolerances that apply to the part being welded. The Unicode weld annotation described in Paper II is modified to incorporate the groove tolerances shown in Fig. 5.8.

The corresponding entities capturing the weld seam geometry and weld annotations are shown in Fig. 5.9. The tolerance information, in combination with seam geometry, will give the allowed deviation in the geometry and groove size. This information can be used as the starting input for the vision-based seam tracking systems.

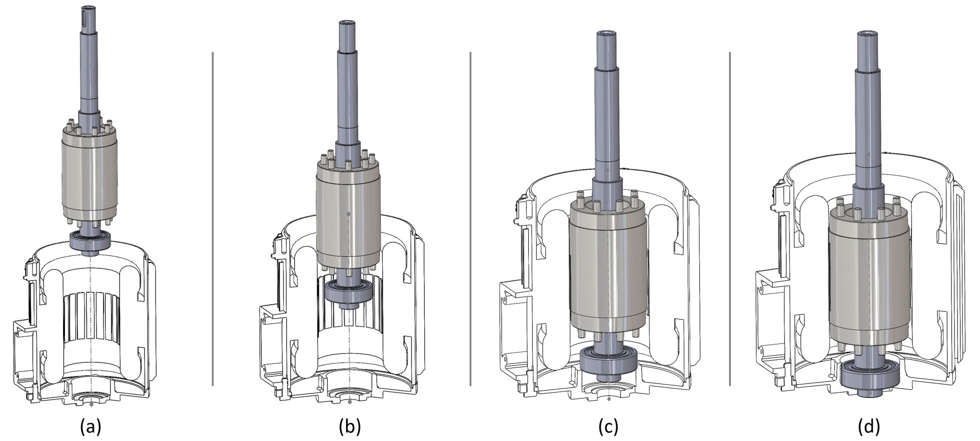


Figure 5.7.: Rotor – Housing Assembly Stages: (a) Initial approach (b) First stage (c) Second stage (d) Final stage

5.4.1. Contributions of Paper III

This paper expands the utility of PMI by seamlessly incorporating GD&T information in STEP AP242 files for reuse in robotic assembly and welding operations. Both methods, the simple Unicode strings and the use of standard entities, offer semantic richness and pave the way for more integrated and automated robotic manufacturing. Below are the key contributions from this paper.

- The GD&T annotations are added semantically to the STEP AP242 files employing Unicode strings and standard entities from the STEP standard.
- This paper introduces a comprehensive method for the extraction of GD&T data in conjunction with associated mating features.
- The GD&T information is extracted and interpreted to calculate the clearances between the mating features. This information is pivotal in estimating assembly forces and determining the appropriate control strategy.
- Modifications have been made to the Unicode weld annotation string described in Paper II to encompass groove tolerances. When applied to the groove or edge geometry, these tolerances influence parameters such as the root gap, bevel angle, and root face. The refined weld annotation delineates the weld seam, its nominal geometry, and the groove geometry. Such detailed information about the seam, potential deviations in its geometry, and edge geometry is instrumental in strategizing scanning and seam tracking activities, particularly when employing visual sensors.

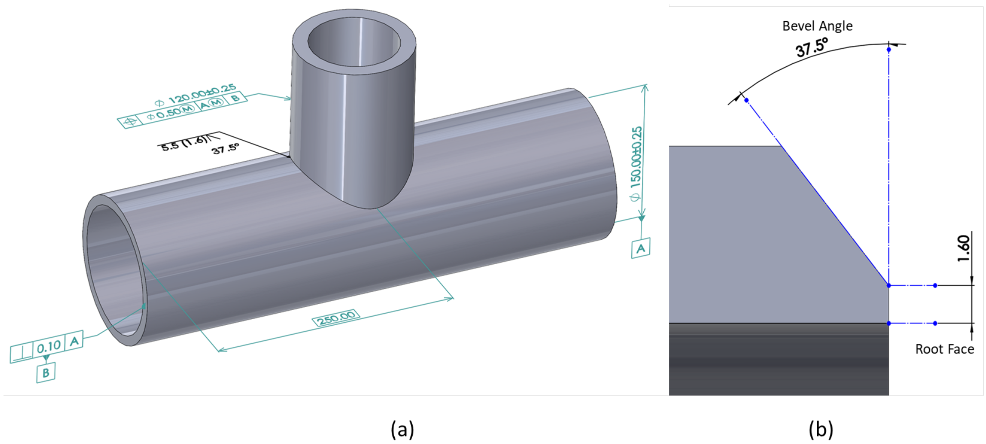
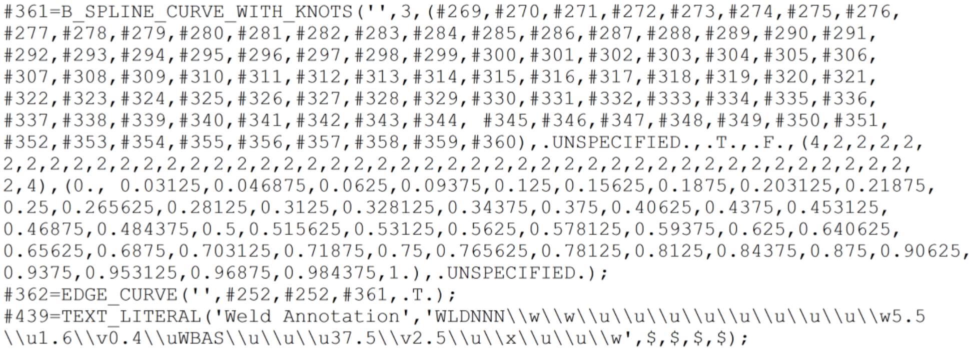


Figure 5.8.: (a) T-joint of Pipes with GD&T and Weld Annotations (b) Groove Geometry

5.5. Summary of Paper IV

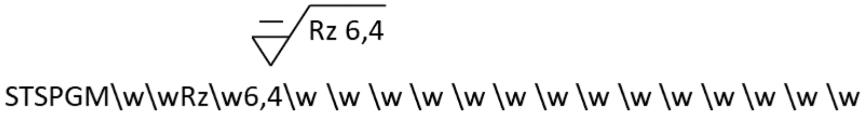
This paper thoroughly explores the advanced utilization of PMI derived from STEP AP242 neutral files. This utilization is tailored explicitly for the optimization of gripper selection and grasp planning in the domain of robotic assembly operations. Drawing from the prior research, where a method to semantically embed welding and GD&T information into STEP files was proposed, this paper augments this methodology to assimilate surface finish annotations and other PMI into the STEP AP242 Ed2 files (**RQ3**). These enriched STEP files subsequently serve as a foundation for delineating constraints for grasp planning (**RQ4** and **RQ5**). Furthermore, the paper also outlines the information necessary for gripper selection and grasp planning, drawing from existing literature (**RQ1**).

Similar to earlier papers, this describes two ways of adding the surface finish information to the STEP AP242 files, one using Custom annotation strings and the other using standard entities. The addition of surface finish information to the product designs is guided by *ISO 21920-1:2021-Geometrical product specifications (GPS)—Surface texture: Profile—Part 1: Indication of surface texture* [56] for profile method and *ISO 25178-1:2016-Geometrical product specifications (GPS)—Surface texture: Areal—Part 1: Indication of surface texture* [57] for areal method. An exhaustive exposition of the surface texture symbols and their intrinsic parameters from these and earlier (now obsolete) ISO standards is proffered. The standard for profile method *ISO 21920-1: 2021* is the latest one that replaced the earlier standard *ISO 1302:2002-Geometrical Product Specifications (GPS)—Indication of surface texture in technical product documentation* [54].



```
#361=B_SPLINE_CURVE_WITH_KNOTS('',3,(#269,#270,#271,#272,#273,#274,#275,#276,
#277,#278,#279,#280,#281,#282,#283,#284,#285,#286,#287,#288,#289,#290,#291,
#292,#293,#294,#295,#296,#297,#298,#299,#300,#301,#302,#303,#304,#305,#306,
#307,#308,#309,#310,#311,#312,#313,#314,#315,#316,#317,#318,#319,#320,#321,
#322,#323,#324,#325,#326,#327,#328,#329,#330,#331,#332,#333,#334,#335,#336,
#337,#338,#339,#340,#341,#342,#343,#344, #345,#346,#347,#348,#349,#350,#351,
#352,#353,#354,#355,#356,#357,#358,#359,#360),.UNSPECIFIED.,.T.,.F.,(4,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,4),(0., 0.03125,0.046875,0.0625,0.09375,0.125,0.15625,0.1875,0.203125,0.21875,
0.25,0.265625,0.28125,0.3125,0.328125,0.34375,0.375,0.40625,0.4375,0.453125,
0.46875,0.484375,0.5,0.515625,0.53125,0.5625,0.578125,0.59375,0.625,0.640625,
0.65625,0.6875,0.703125,0.71875,0.75,0.765625,0.78125,0.8125,0.84375,0.875,0.90625,
0.9375,0.953125,0.96875,0.984375,1.),.UNSPECIFIED.);
#362=EDGE_CURVE('',#252,#252,#361,.T.);
#439=TEXT_LITERAL('Weld Annotation','WLDNNN\\w\\w\\u\\u\\u\\u\\u\\u\\u\\u\\w5.5
\\u1.6\\v0.4\\uWBAS\\u\\u\\u37.5\\v2.5\\u\\x\\u\\u\\w',$,$,$,$);
```

Figure 5.9.: Portion of STEP File showing the Entities for Weld Seam and Weld Annotations.



STSPGM\w\wRz\w6,4\w \w \w \w \w \w \w \w \w \w \w \w \w \w

Figure 5.10.: Surface texture symbol with R-parameter, tolerance limit value, and the corresponding annotation string.

A method is developed to form a semantic custom annotation string that carries all the surface finish information per the latest ISO standards. This method is simple, and the annotation string is added to the STEP file using the 'TEXT_LITERAL' entity. Fig. 5.10 shows a surface finish symbol and its equivalent custom annotation string.

A second method of adding surface finish symbols using entities defined in the STEP AP242 standard is described. The capabilities to include surface finish information in STEP AP242 are defined for the now-obsolete *ISO 1302:2002-Geometrical Product Specifications (GPS)—Indication of surface texture in technical product documentation* [54] standard. This earlier surface finish standard is replaced by *ISO 21920-1: 2021* standard mentioned earlier. A re-mapping of most of the entities of Application Module *Surface conditions* (Part: 1110) from STEP AP242 standard with the symbols and parameters from the latest *ISO 21920-1: 2021* standard is done. A method to add the surface finish parameters that cannot be matched directly to the standard entities of STEP AP242 is suggested. Fig. 5.11 shows the addition of the 'tolerance acceptance rule' to the STEP file using 'DESCRIPTIVE_REPRESENTATION_ITEM' entity.

Other PMI like thread information, surface treatment/coatings, material, volume,

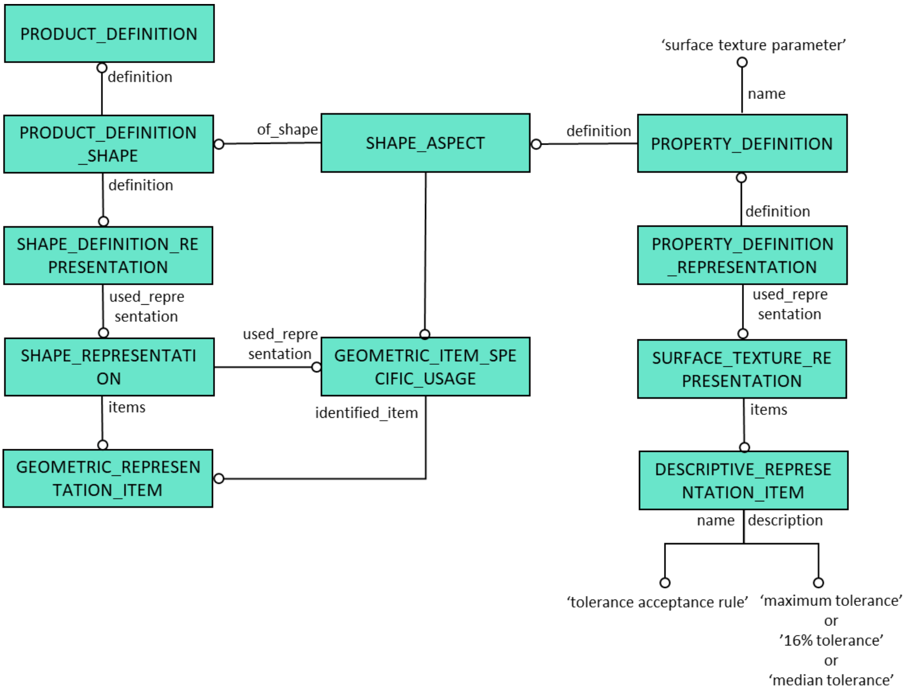


Figure 5.11.: Addition of surface texture parameter—tolerance acceptance rule to STEP AP242 file.

mass, and bounding box that can be required for gripper selection and grasp planning are also discussed. Entities that can carry this kind of PMI are identified from the standard, and the methods to include them in the STEP file are described.

A nuanced classification of handling features into Forbidden, Restricted, and Grasp features is introduced. The criterion and the method to identify those features from the PMI extracted from STEP files are presented. The various grasp features of the Rotor from the Motor assembly are shown in Fig. 5.12.

A two-level process is suggested to select the appropriate grippers using the product information. The first level takes into consideration the geometric properties of the grasp feature and part weight to narrow down the grippers with the required grasp width, stroke, and payload. In the second level, gripping forces are considered to select the appropriate grippers. This process is successfully applied to shortlist five suitable grippers for handling the Rotor from a total of 24 grippers from four different manufacturers.

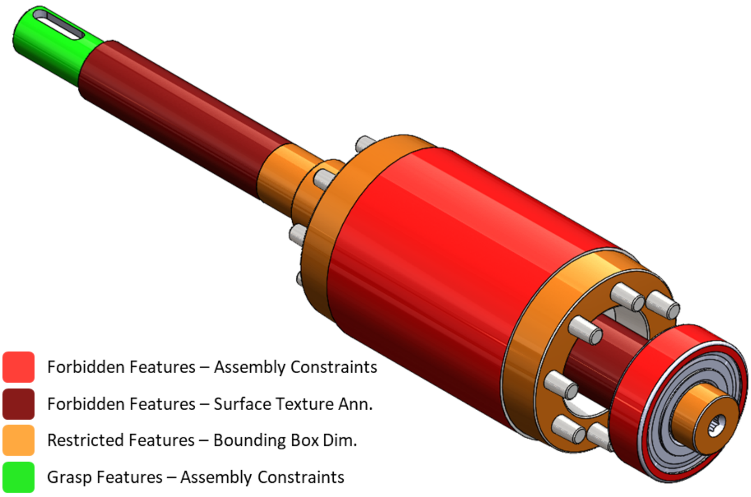

🟥 Forbidden Features – Assembly Constraints
🟫 Forbidden Features – Surface Texture Ann.
🟧 Restricted Features – Bounding Box Dim.
🟩 Grasp Features – Assembly Constraints

Figure 5.12.: Handling features of Rotor sub-assembly.

5.5.1. Contributions of Paper IV

The addition, extraction, and possible uses of other PMI like surface finish, hole annotations, thread information, material, and volume are discussed in this paper. The current international standards concerning the surface finish information are analyzed and compared with the capabilities of the STEP AP242 standard. The methods of adding surface finish annotations are made compatible with the latest applicable standards. The contributions of this paper are

- Described methods for the semantic integration of PMI elements, such as surface finish, thread information, and material properties, into the STEP AP242 Ed2 files.
- The surface texture annotations have been incorporated in strict adherence to the ISO 21920-1:2021 standard. This is noteworthy, given that the STEP AP242 standard was originally defined in alignment with the now-obsolete ISO 1302:2002.
- Introduced comprehensive classification system for handling features. Furthermore, it proposed a method to identify them using the product information from the STEP files
- Presented a two-level approach for selecting appropriate grippers utilizing the rich product information embedded within the STEP files.

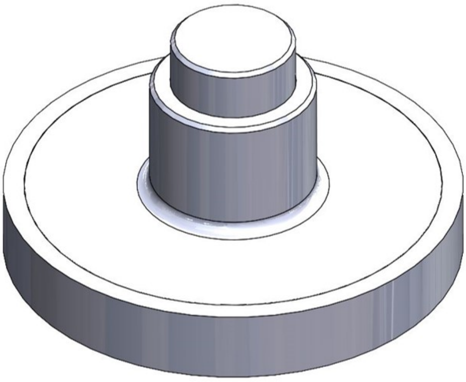


Figure 5.13.: Peg-Hole Assembly Test Case with Two-Stage Insertion

5.6. Summary of Paper V

This paper builds on the work done till now in this thesis and explores the integration of product data from STEP AP242 files into robotic assembly operations. This paper validates the approach of this study in creating and reusing the product data from semantic MBD files by establishing a Digital Thread for the continuous flow of product data (**RQ5**), thus bridging the gap between the design phase and robotic assembly. This integration of product data into robotic programming is manifested through various assembly experiments using two test cases executed using the Franka Emika robotic arm.

The first test case employs the inverted peg-hole assembly with a two-stage insertion shown in Fig. 5.13. This straightforward scenario serves as a foundation to demonstrate data transfer from STEP files to the robotic assembly program, paving the way for more complex test cases.

The Cranfield assembly benchmark, illustrated in Fig. 5.14, is selected as the second test case. This acknowledged benchmark encompasses five elementary peg-hole insertions and three complex insertion tasks, presenting variations in component sizes and shapes. From a CAD model perspective, the benchmark introduces ample complexity in the geometry of the parts and the assembly structure, with dual instances of round pegs, square pegs, and both top and bottom plates. Given these attributes, the benchmark aptly assesses the potential of directly leveraging product data from STEP AP242 files for robotic automation.

The PMI is added to the STEP AP242 files semantically by employing the methodology described in previous papers. The PMI and geometric details from STEP AP242 files are extracted. The following product data is extracted from the STEP

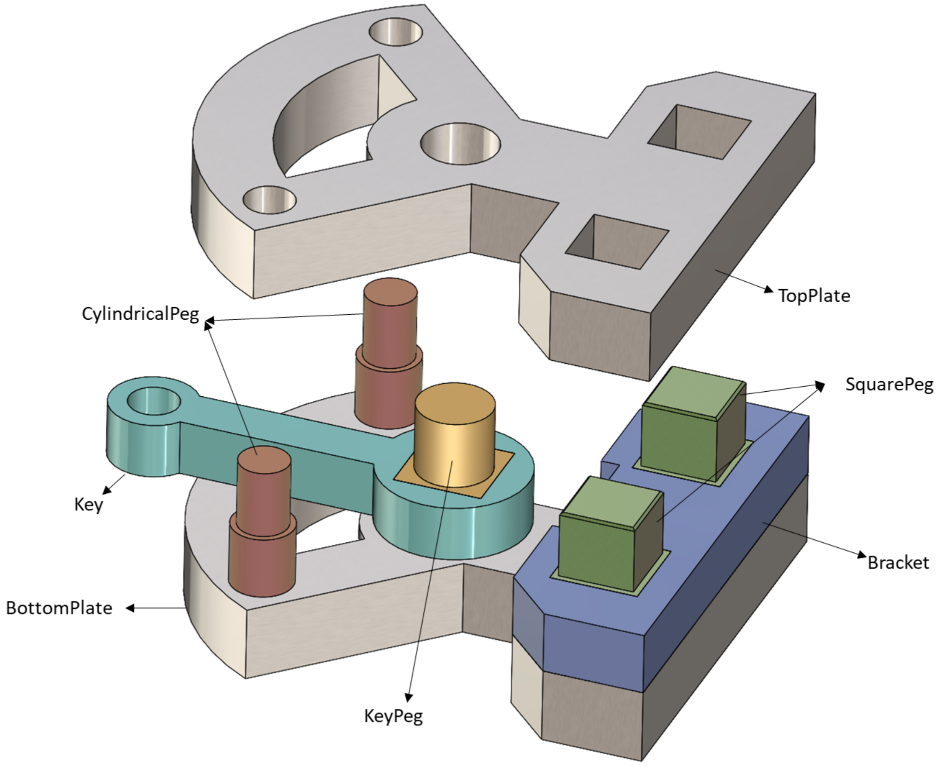
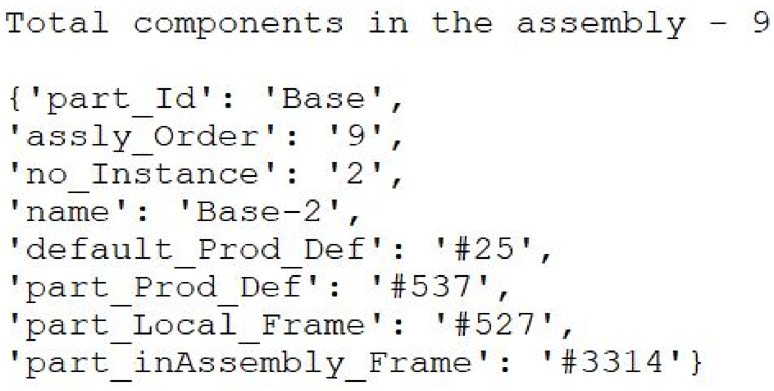


Figure 5.14.: Cranfield Assembly Benchmark

files.

- Assembly coordinate system
- Total number of components in the assembly
- The order in which the components were added to the assembly (assumed to be the assembly order)
- Part coordinate system with respect to the assembly coordinate system
- Assembly constraints on the part and mating surfaces
- Coordinate system of the mating surfaces with respect to the part coordinate system
- Clearance between the mating parts
- Grasp features and their sizes



```
Total components in the assembly - 9

{'part_Id': 'Base',
'assly_Order': '9',
'no_Instance': '2',
'name': 'Base-2',
'default_Prod_Def': '#25',
'part_Prod_Def': '#537',
'part_Local_Frame': '#527',
'part_inAssembly_Frame': '#3314'}
```

Figure 5.15.: Component details of TopPlate extracted from the STEP file

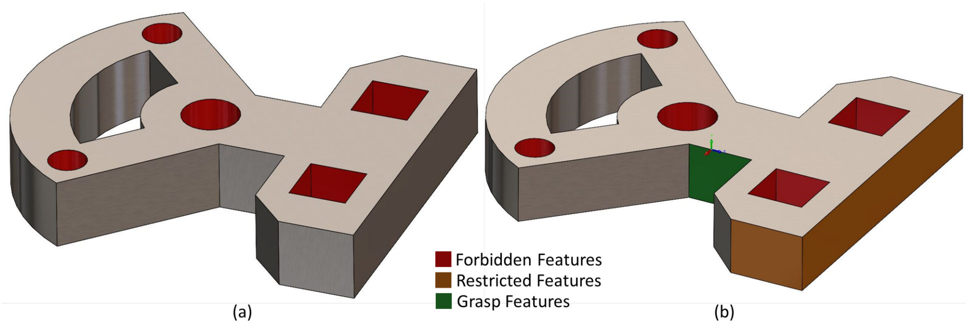


Figure 5.16.: Top Plate: Handling Features from (a) Earlier Criteria; (b) Adjusted Criteria

- Coordinate system of the grasp surfaces with respect to the part coordinate system

This data also includes the number of instances of a particular part used in that assembly, as shown in Fig. 5.15 for the second instances of part_Id ‘Base,’ which is used for both ‘BottomPlate’ and ‘TopPlate’ in the assembly.

This extracted data plays a pivotal role in defining robotic assembly tasks, such as defining spatial relationships between components, selecting appropriate grippers, and planning grasps. The Cranfield assembly benchmark presented two scenarios, necessitating the adjustment to the process of identifying the grasp features described in Pape IV.

- For the Top Plate, in the absence of other annotations like surface finish, threading, or painting/coatings on the surfaces, all other surfaces except

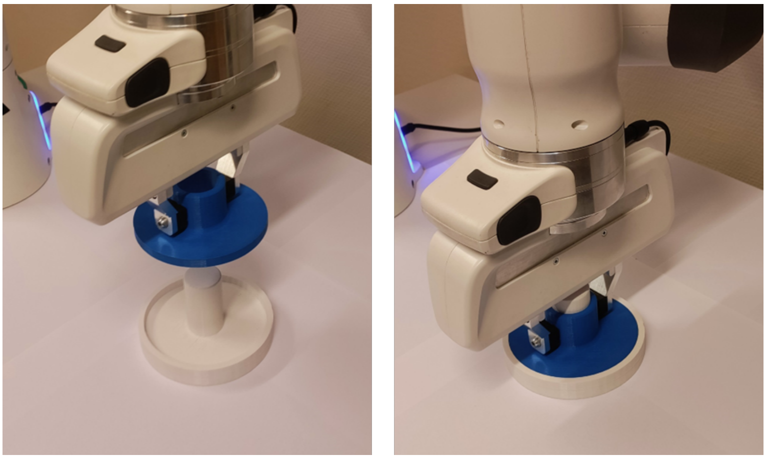


Figure 5.17.: Experiments with Inverted Peg-Hole Assembly

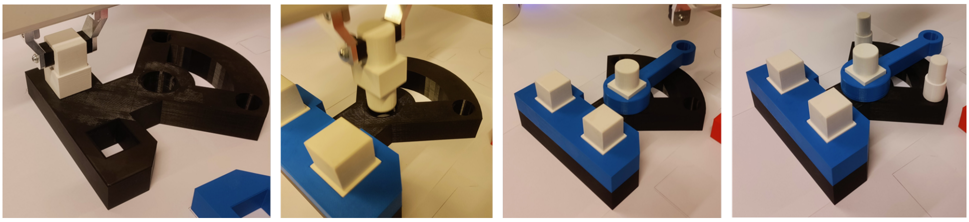


Figure 5.18.: Experiments with Cranfield assembly benchmark

the one with assembly constraints become eligible candidates for grasp features. The following conditions are applied to select the appropriate Grasp Features.

1. Only parallel surfaces are considered
2. Closeness to the center of gravity

Fig. 5.16 shows the handling features identified by following the earlier and updated criteria.

- The bounding box height of the Key and the Bracket are less than the insertion lengths. In this case, the condition of the bounding box is suspended, and the rest of the process is applied.

To validate the hypothesis of this thesis, various assembly experiments were conducted on the 3D printed models of the peg-hole assembly test case, shown in Fig. 5.17 and Cranfield assembly benchmark shown in Fig. 5.18.

These experiments ascertained the efficacy of STEP AP242 files as reliable carriers of product data for robotic automation. The results of these experiments were promising, underscoring the potential of the proposed methodology in facilitating a connected Digital Thread for robotic assembly. The successful extraction of product data and the completion of several assembly tasks demonstrated the viability of using STEP files for robot programming. However, closely toleranced parts need a more precise and targeted control strategy.

5.6.1. Contributions of Paper V

This paper delved into the realization of product data integration from STEP AP242 files into robotic assembly operations. The efficacy of extracting and reusing geometrical data and PMI for various assembly tasks was validated through experiments. The contributions of this paper are

- Demonstrated how PMI, combined with geometric information from STEP AP242 files, can be instrumental in various facets of robotic assembly, including defining spatial relationships between components, aiding in gripper selection, and facilitating grasp planning.
- Validated the utility of product data extracted from STEP AP242 files for robotic assembly operations Through assembly test cases.
- Established a Digital Thread using STEP AP242 neutral files, which ensures a seamless flow of data from the design phase of the product life cycle to robotic assembly.

Chapter 6.

Contributions

The main contribution of this thesis is the establishment of a Digital Thread for robotic assembly from the MBD created during the design phase using the STEP AP242 standard. The semantic inclusion of product data in the MBD using STEP AP242 enables the flow of product data from the design to robotic manufacturing. The thesis focuses on creating a semantic MBD so that the product data can be transferred to the assembly operations and reused for automated robot programming with minimal manual recreation of data. The approach presented in this thesis and the research papers covers all the aspects of creating a semantic MBD that captures most of the PMI that is generally included in the 2D manufacturing drawing. The PMI is added as per the international ISO and ASME standards. Another important aspect is that all the added PMI is semantic, even though some of these standards do not deal with semantic PMI. Additional PMI not described in the papers can be included in the STEP AP242 files by following a similar method. Another benefit of this approach is using STEP files, which makes this method CAD independent. Using STEP neutral exchange files increases the ease of adaptation of digitization across the industry. It relieves the SMEs from the burden of purchasing CAD licenses to access semantic MBD.

Another aspect of the thesis' contributions is the automated programming of industrial robots using a constraint-based methodology. The assembly constraints, mating feature geometry, the GD&T, welding, and other PMI are used to identify various constraints for robotic task specification. As the PMI is semantically added to the CAD models during the design phase, the manual extraction and recreation of this information in the robot programming phase are reduced significantly. The extracted product data is used in the robot programming as demonstrated through the assembly experiments.

Below are some of the key contributions of this thesis.

- Formation of a seamless channel for product data for robotic automation using MBD

- Establishing the STEP AP242 standard as the carrier of product design data for the realization of the digital thread in robotic assembly
- Semantic encapsulation of all the product data needed for the programming of the assembly robot in the STEP files
- Simple yet efficient method of forming custom Unicode strings to overcome the lack of schema and entity definitions for the addition of PMI in STEP files
- Adherence to the latest international standards and industry practices while creating MBD
- Devised methods to extract the most relevant information from the STEP files like spatial relationships between the components, mating features, clearances between the mating features, seam geometry, grasp features
- Introduction of a fresh classification of handling features specifically for robotic assembly
- Proposed a criterion for the selection of suitable grippers utilizing the product data
- Realization of robotic assembly programming using product data from STEP AP242 files

Chapter 7.

Discussion

In addressing the research questions posed, this study has embarked on a comprehensive exploration of the integration of product data from the design phase into robotic assembly operations. The following discussion provides a detailed analysis of the work undertaken in this research.

7.1. Research Development

This research started with an investigation into the current state of robot programming for assembly and the potential for reusing product data. It was evident from the literature that part geometry has been consistently utilized since the inception of CAD systems. However, another essential information, the spatial relationship between the constituent parts of the assembly, is being recreated manually for robot programming. This observation prompted the first research question, RQ1, which delved into the specific categories of product data essential for robotic assembly programming.

The exploration into RQ1 unveiled the pivotal role of product information in robotic automation, as detailed in Chapter 2 and the introductory sections of the papers. Recognizing that this information is typically generated during the design phase and traditionally documented in 2D manufacturing drawings, the study mapped the data required during robotic assembly to the product information captured during design.

It is essential to mention that certain information pivotal for the successful execution of the assembly operation is not directly generated/documented with product designs. For example, the assembly sequence is not recorded with the product design but recreated and documented separately in an 'Assembly Instructions' manual, along with the other details like special tools and fixtures needed to complete the assembly. This research has not delved into these external data sources. As stated in Paper I, it was assumed that the assembly sequence is the same as the order in which part models are added to the assembly model in the CAD system during the design.

With the shift towards digitalization and the growing adoption of MBD, it was hypothesized that MBD with rich semantic data could be used as a data source for programming assembly robots, avoiding manual recreation from 2D drawings. This led to the second research question, RQ2, which sought to identify the most appropriate file format for conveying semantic product data. After a thorough evaluation, the STEP AP242 neutral file format emerged as the most fitting choice, meeting the initial criteria described in the last section of Chapter 3.

Subsequently, a meticulous analysis of the STEP AP242 standard was conducted to address RQ3, which delves into the limitations of the STEP AP242 standard. The challenges and potential solutions associated with STEP file creation and interaction are discussed in Chapter 4, along with the alternate solutions used in this thesis.

While the standard effectively conveys geometrical and topological information, its representation of other product data varies. The research adopted a detailed methodology, as outlined in Chapter 5, to evaluate and overcome the limitations in the standard concerning the addition of PMI. The reliance on text annotations and custom Unicode strings as makeshift solutions underscores the need for a more holistic approach to data representation in future editions of the STEP AP242 standard.

Another way to explore the STEP AP242 standard is to flip the third research question: the information-carrying capabilities available with STEP but not used in this thesis for robotic automation. One example of such a capability is the kinematic information, which is used to capture the designs of mechanisms. It would be interesting to explore its potential applications in robotic assembly.

Once the STEP file is semantically embedded with all the required product data, the next step is to find the answers to the fourth research question, RQ4, centered on extracting and interpreting product data from MBD files for robotic assembly. A systematic process was developed, as detailed in Paper I, to discern the relative positions of assembly components and their spatial relationships. Additional methods were developed to extract the critical parameters of mating/grasp features like clearances and dimensions. Furthermore, parsers were created to interpret PMI from semantic Unicode strings, converting them into robotic task parameters. The extraction process was streamlined to cater specifically to the automation of assembly tasks.

The coding, simulation, and experiments, as presented in Papers I-V, validate the efficacy of the developed methods. The components used in the experiments underscore the versatility of the process across various part shapes and sizes. However, there remains a need to refine control strategies for the assembly of components with tight tolerances.

In summary, this research has contributed to robotic assembly by harnessing product data from the design phase. By identifying the essential data categories, selecting and refining suitable MBD file formats, and developing methodologies for data extraction, the study has laid a solid foundation for direct reuse of product data for automated assembly operations. The programs and assembly experiments stand as a testament to the practicality of the research methodologies.

Chapter 8.

Conclusion

This thesis has described an approach for using product information from CAD files by establishing a Digital Thread (DT) for automated robot programming for assembly operations. Semantic MBD created using the STEP AP242 standard established the information flow from design to robotic assembly. The methodology presented in this thesis enables the reuse of product information, both geometric and PMI, for robotic automation without manual recreation of product data. This thesis also establishes the STEP AP242 files as the carriers and source of product definition for downstream operations. It is also demonstrated that all the information traditionally included in the 2D manufacturing drawings can be included in the STEP AP242 files. Using neutral STEP exchange files facilitates the creation of an unbroken chain of product information from design to robotic manufacturing throughout the extended enterprise.

The published papers demonstrated that the product information from STEP AP242 files could be extracted and reused for constraint specification for automated robot programming for assembly operations. The mating features and the relative positions of the parts in the assembly were extracted directly from the STEP AP242 files without human involvement. These spatial relations were used to specify the distance measure in a constraint-based robot programming framework to demonstrate the robotic assembly of a motor. The welding and GD&T annotations from STEP files were used to define the critical product and process parameters for welding and assembly operations. The weld seam geometry, along with the welding process information, can be used in defining the constraints on a welding robot. The GD&T information was used to calculate the clearances between the mating components, which can be used to estimate the assembly forces and decide the control methodology. Combining the welding information with GD&T, the possible deviation in weld seam and edge geometry can be estimated, which then can be used for defining the seam tracking operations using visual sensing.

PMI-enriched STEP files were used to specify the constraints critical for gripper selection and grasp planning. A detailed classification of the handling features suitable for robotic assembly was introduced. A method to identify appropriate handling features is presented and demonstrated with a use case. Criteria for gripper selection were presented, and appropriate grippers were selected from the surface finish annotations and other PMI for the identified grasp features. The product information from STEP AP242 files was used for robot programming of assembly tasks. The methods' applicability was demonstrated through assembly tests that demonstrate their suitability for various sizes and shapes of the assembled objects. These experiments corroborate the central hypothesis of this research, thereby affirming the formation of a seamless digital thread facilitating the transfer of product data from design to robotic assembly using STEP AP242 files.

8.1. Future Work

There are several possibilities for exciting and valuable research related to assembly information and automation. This section lists some aspects of assembly information and STEP that are not covered under this thesis, which would be a good starting point for further research. This section also mentions some of the directions in which STEP adaptation can be increased in robotic automation.

- Exploring the kinematic capabilities of STEP AP242 for multi-robot assembly is a logical progression. After each assembly step, the mating components can be analyzed for available degrees of freedom, and the robot manipulator can be constrained to avoid such directions during the next stages of assembly. The manipulator kinematics can be extended to include the part as another link of the manipulator as suggested by Whitney [22].
- Another promising avenue is the utilization of feature normals of the mating features to determine the assembly direction and sequence. The screw theory, particularly the concept of repelling and contrary screws [89] can be used to find the assembly and disassembly directions [22].
- The integration of STEP and robot programming can be increased by adopting a description of the robotic manipulator in terms of STEP standard. STEP has the scope to capture the geometry of the linkages and the kinematics of the joints. A Robot Description Format based on STEP, for example, STEP-RDF, can be developed to describe the robots.
- One valuable work would be the extension of the STEP standards (STEP-Robo) to include robot descriptions and applications similar to modules of STEP-NC.

Bibliography

[1] A. P. Ambler, Harry G. Barrow, Christopher M. Brown, Rod M. Burstall, Robin J. Popplestone. "A Versatile System for Conputer-Controlled Assembly". In: *Artificial Intelligence* 6 (1975), pp. 129–156.

[2] A. P. Ambler, Robin J. Popplestone. "Inferring the Positions of Bodies from Specified Spatial Relationships". In: *Artificial Intelligence* 6 (1975), pp. 157–174.

[3] Alexander M. Miller, Nathan W. Hartman, Thomas Hedberg, Allison B. Feeney, Jesse Zahner. "Towards Identifying the Elements of a Minimum Information Model for use in a Model-Based Definition". In: *Proceedings of the ASME 2017 12th International Manufacturing Science and Engineering Conference*. Vol. Volume 3: Manufacturing Equipment and Systems. International Manufacturing Science and Engineering Conference. 2017.

[4] Alexander Perzylo, Markus Rickert, Björn Kahl, Nikhil Somani, Christian Lehmann, Alexander Kuss, Stefan Profanter, Anders B. Beck, Mathias Haage, Mikkel R. Hansen, Malene T. Nibe, Máximo A. Roa, Olof Sörnmo, Sven G. Robertz, Ulrike Thomas, Germano Veiga, Elin A. Topp, Ingmar Kessler, Marinus Danzer. "SMErobotics: Smart Robots for Flexible Manufacturing". In: *IEEE Robotics Automation Magazine* 26.1 (2019), pp. 78–90.

[5] Alexander Perzylo, Nikhil Somani, Stefan Profanter, Ingmar Kessler, Markus Rickert, Alois Knoll. "Intuitive instruction of industrial robots: Semantic process descriptions for small lot production". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2016, pp. 2293–2300.

[6] Allison B. Feeney. "The STEP Modular Architecture". In: *ASME Journal of Computing and Information Science in Engineering* 2 (2002), pp. 132–135.

[7] Allison B. Feeney, Simon P. Frechette, Vijay Srinivasan. "A Portrait of an ISO STEP Tolerancing Standard as an Enabler of Smart Manufacturing Systems". In: *ASME Journal of Computing and Information Science in Engineering* 15.2 (2015), pp. 021001–021001-5.

[8] ASD Strategic Standardisation Group. “Through Life Cycle Interoperability - A critical strategic lever for competitiveness”. In: Rev. 1.0 (2014). URL: %7Bhttp://www.asd-ssg.org/through-life-cycle-interoperability%7D.

[9] ASME Y14.41:2003. “Digital Product Definition Data Practices”. In: *The American Society of Mechanical Engineers* (2003).

[10] ASME Y14.41:2019. “Digital Product Definition Data Practices”. In: *The American Society of Mechanical Engineers* (2019).

[11] ASME Y14.5:2018. “Dimensioning and Tolerancing”. In: *The American Society of Mechanical Engineers* ().

[12] Aude Billard, Sylvain Calinon, Rudiger Dillmann, Stefan Schaal. “Robot Programming by Demonstration”. In: *Springer Handbook of Robotics*. Springer-Verlag Berlin Heidelberg, 2008, pp. 1371–1394. ISBN: 978-3-540-23957-4.

[13] Marnix van Bruggen, Jan Peter Baartman, and Willem F. Bronsvoort. “Grips on Parts”. In: *Proceedings IEEE International Conference on Robotics and Automation*. Vol. 2. 1993, pp. 828–833.

[14] Bruno Lotter. *Manufacturing Assembly Handbook*. VDI-Verlag GmbH, 1986. ISBN: 0-408-03561-7.

[15] H. Bruyninckx, S. Dutre, and J. De Schutter. “Peg-on-Hole: A Model Based Solution to Peg and Hole Alignment”. In: *IEEE International Conference on Robotics and Automation*. 1995, pp. 1919–1924.

[16] Carlos Mateo, Alberto Brunete, Ernesto Gambao, Miguel Hernando. “Hammer: An Android Based Application for End-User Industrial Robot Programming”. In: *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. 2014, pp. 1–6.

[17] CAx Implementor Forum. “CAx-IF Recommended Practices for the Representation and Presentation of Product Manufacturing Information (PMI) (AP242)”. In: Version 4.0 (2014).

[18] CAx Implementor Forum. “PMI Unicode String Specification Examples and Mapping Strategies - for Dimensioning and Tolerancing, GD&T, Surface Texture Symbol, and Welding Symbol PMI Annotation Entities Revision”. In: Revision J (2011).

[19] Seong-Youb Chung and D. Y. Lee. “Discrete Event Systems Approach to Fixtureless Peg-in-Hole Assembly”. In: *Proceedings of the American Control Conference*. 2001, pp. 4962–4967.

[20] Colin H. Simmons, Dennis E. Maguire, Neil Phelps. *Manual of Engineering Drawing*. Butterworth-Heinemann, 2020. ISBN: 978-0-12-818482-0.

[21] Cuadrado J. and Naya M.A. and Ceccarelli M. and Carbone G. “An optimum design procedure for two-finger grippers: a case of study”. In: *IFToMM Electronic Journal of Computational Kinematics* 15403.1 (2002).

[22] Daniel E. Whitney. *Mechanical Assemblies: Their Design, Manufacturing, and Role in Product Development.* Oxford University Press, 2004. ISBN: 978-0-19-806022-2.

[23] Department of Defense, United States of America (USA). *MIL-STD-31000A: Technical Data Packages.* Federal Standardization Manual. Department of Defense, USA, 2009.

[24] Department of Defense, United States of America (USA). *MIL-STD-31000B: Technical Data Packages.* Federal Standardization Manual. Department of Defense, USA, 2018.

[25] F. Dietrich, D. Buchholz, F. Wobbe, F. Sowinski, A. Raatz, W. Schumacher, and F. M. Wahl. “On Contact Models for Assembly Tasks: Experimental Investigation Beyond the Peg-in-Hole Problem on the Example of Force-Torque Maps”. In: *The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2010, pp. 2313–2318.

[26] Douglas Schenck, Peter Wilson. *Information Modeling the EXPRESS Way.* Oxford University Press, Inc., 200 Madison Avenue, New York, 1994. ISBN: 9780195087147.

[27] Edited by Julian M. Muller, Nikolai Kazantsev. *Industry 4.0 in SMEs Across the Globe: Drivers, Barriers, and Opportunities.* CRC Press, 2022. ISBN: 978-0-367-76190-5.

[28] H. A. ElMaraghy, W. H. ElMaraghy, and L. Knoll. “Design Specification of Parts Dimensional Tolerance for Robotic Assembly”. In: *Computers in Industry* 10 (1988), pp. 47–59.

[29] Erwin Aertbelien, Joris De Schutter. “eTaSL/eTC: A constraint-based Task Specification Language and Robot Controller using Expression Graphs”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2014), pp. 1540–1546.

[30] European Commission. *Industry 5.0: Towards a sustainable, human centric and resilient European industry.* European Commission, 2021.

[31] European Commission. *SME definition.* URL: https://ec.europa.eu/growth/smes/sme-definition_en. (accessed: 29.05.2022).

[32] Gualtiero Fantoni, Saverio Capiferri, and Jacopo Tilli. “Method for supporting the selection of robot grippers”. In: *Procedia CIRP* 21 (2014), pp. 330–335.

[33] Geoffrey Biggs, Bruce McDonald. “A Survey of Robot Programming Systems”. In: *Proceedings of the Australasian Conference on Robotics and Automation*. 2003, pp. 1–3.

[34] Gopika Ajaykumar, Maureen Steele, Chien-Ming Huang. “A Survey on End-Use Robot Programming”. In: *ACM Computing Surveys* 54.8 (2021), Article 164.

[35] Gregory A. Harris, Chris Peters, Ashley Yarbrough, Cole Estes, Daniel Abernathy. “Industry Readiness for Digital Manufacturing May Not Be as We Thought Preliminary Findings of MxD Project 17-01-01”. In: *Proc. of the 10th Model-Based Enterprise Summit (MBE2019)*. 2019, pp. 110–116.

[36] Hans P. Moravec. *robot*. 2021. URL: https://www.britannica.com/technology/robot-technology. (accessed: 20.05.2022).

[37] W. Haskiya, K. Maycock, and J. Knight. “Robotic assembly: chamferless peg-hole assembly”. In: *Robotics* 17 (1999), pp. 621–634.

[38] Heiko Mosemann, Friedrich M. Wahl. “Automatic Decomposition of Planned Assembly Sequences Into Skill Primitives”. In: *IEEE Transactions on Robotics and Automation* 17.5 (2001), pp. 709–718.

[39] Henning Kagermann, Wolfgang Wahlster, Johannes Helbig. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry; Final Report of the Industrie 4.0 Working Group*. Forschungsunion, 2013.

[40] M. Honarpardaz, M. Tarkian, J. Olvander, and X. Feng. “Finger design automation for industrial robot grippers: A review”. In: *Robotics and Autonomous Systems* 87 (2017), pp. 104–119.

[41] ISO 10303-1:2021. “Industrial automation systems and integration — Product data representation and exchange - Part 1: Overview and fundamental principles”. In: *International Organization for Standardization, Geneva, Switzerland* (2021).

[42] ISO 10303-11:2004. “Industrial automation systems and integration — Product data representation and exchange - Part 11: Description methods: The EXPRESS language reference manual”. In: *International Organization for Standardization, Geneva, Switzerland* (2004).

[43] ISO 10303-21:2016. “Industrial automation systems and integration — Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure”. In: *International Organization for Standardization, Geneva, Switzerland* (2016).

[44] ISO 10303-22:1998. “Industrial automation systems and integration — Product data representation and exchange - Part 22: Implementation methods: Standard data access interface”. In: *International Organization for Standardization, Geneva, Switzerland* (1998).

[45] ISO 10303-23:2000. “Industrial automation systems and integration — Product data representation and exchange - Part 23: Implementation methods: C++ language binding to the standard data access interface”. In: *International Organization for Standardization, Geneva, Switzerland* (2000).

[46] ISO 10303-24:2001. “Industrial automation systems and integration — Product data representation and exchange - Part 24: Implementation methods: C language binding of standard data access interface”. In: *International Organization for Standardization, Geneva, Switzerland* (2001).

[47] ISO 10303-242:2014. “Industrial automation systems and integration — Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering”. In: *International Organization for Standardization, Geneva, Switzerland* (2014).

[48] ISO 10303-242:2020. “Industrial automation systems and integration — Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering”. In: *International Organization for Standardization, Geneva, Switzerland* (2020).

[49] ISO 10303-28:2007. “Industrial automation systems and integration — Product data representation and exchange - Part 27: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas”. In: *International Organization for Standardization, Geneva, Switzerland* (2007).

[50] ISO 10303-31:1994. “Industrial automation systems and integration — Product data representation and exchange - Part 31: Conformance testing methodology and framework: General concepts”. In: *International Organization for Standardization, Geneva, Switzerland* (1994).

[51] ISO 10303-32:1998. “Industrial automation systems and integration — Product data representation and exchange - Part 32: Conformance testing methodology and framework: Requirements on testing laboratories and clients”. In: *International Organization for Standardization, Geneva, Switzerland* (1998).

[52] ISO 10303-34:2001. “Industrial automation systems and integration — Product data representation and exchange - Part 34: Conformance testing methodology and framework: Abstract test methods for application protocol implementations”. In: *International Organization for Standardization, Geneva, Switzerland* (2001).

[53] ISO 1101:2017. “Geometrical product specifications (GPS) — Geometrical tolerancing — Tolerances of form, orientation, location and run-out”. In: *International Organization for Standardization, Geneva, Switzerland* ().

[54] ISO 1302:2002. “Geometrical Product Specifications (GPS) — Indication of surface texture in technical product documentation”. In: *International Organization for Standardization, Geneva, Switzerland* ().

[55] ISO 14306:2017. “Industrial automation systems and integration — JT file format specification for 3D visualization”. In: *International Organization for Standardization, Geneva, Switzerland* (2017).

[56] ISO 21920-1:2021. “Geometrical product specifications (GPS) — Surface texture: Profile — Part 1: Indication of surface texture”. In: *International Organization for Standardization, Geneva, Switzerland* ().

[57] ISO 25178-1:2016. “Geometrical product specifications (GPS) — Surface texture: Areal — Part 1: Indication of surface texture”. In: *International Organization for Standardization, Geneva, Switzerland* ().

[58] ISO/TC 184/SC 4/WG 12 N10645. “Industrial automation systems and integration — Product data representation and exchange — Progress towards STEP Part 4000 Core Model”. In: *International Organization for Standardization, Geneva, Switzerland* (2020).

[59] ISO/TS 10303-25:2005. “Industrial automation systems and integration — Product data representation and exchange - Part 25: Implementation methods: EXPRESS to XMI binding”. In: *International Organization for Standardization, Geneva, Switzerland* (2005).

[60] ISO/TS 10303-26:2011. “Industrial automation systems and integration — Product data representation and exchange - Part 26: Implementation methods: Binary representation of EXPRESS-driven data”. In: *International Organization for Standardization, Geneva, Switzerland* (2011).

[61] ISO/TS 10303-27:2000. “Industrial automation systems and integration — Product data representation and exchange - Part 27: Implementation methods: Java TM programming language binding to the standard data access interface with Internet/Intranet extensions”. In: *International Organization for Standardization, Geneva, Switzerland* (2000).

[62] ISO/TS 10303-304:2001. “Industrial automation systems and integration — Product data representation and exchange - Part 304: Abstract test suite: Mechanical design using boundary representation”. In: *International Organization for Standardization, Geneva, Switzerland* (2001).

[63] ISO/TS 10303-307:2000. “Industrial automation systems and integration — Product data representation and exchange - Part 307: Abstract test suite: Sheet metal die planning and design”. In: *International Organization for Standardization, Geneva, Switzerland* (2000).

[64] ISO/TS 10303-325:2004. “Industrial automation systems and integration — Product data representation and exchange - Part 325: Abstract test suite: Building elements using explicit shape representation”. In: *International Organization for Standardization, Geneva, Switzerland* (2004).

[65] ISO/TS 10303-332:2002. “Industrial automation systems and integration — Product data representation and exchange - Part 332: Abstract test suite: Technical data packaging core information and exchange”. In: *International Organization for Standardization, Geneva, Switzerland* (2002).

[66] ISO/TS 10303-35:2003. “Industrial automation systems and integration — Product data representation and exchange - Part 35: Conformance testing methodology and framework: Abstract test methods for standard data access interface (SDAI) implementations”. In: *International Organization for Standardization, Geneva, Switzerland* (2003).

[67] J. Norberto Pires, Altino Loureiro, Gunnar Bolmsjo. *Welding Robots: Technology, System Issues and Applications*. Springer, 2006. ISBN: 1852339535.

[68] J. Norberto Pires, T. Godinho, P. Ferreira. “CAD Interface for Automatic Robot Welding Programming”. In: *Industrial Robot: An International Journal* 31.1 (2004), pp. 71–76.

[69] Jennifer B. Herron. *Re-Use Your CAD: The Model-Based CAD Handbook*. CreateSpace Independent Publishing Platform, 2013. ISBN: 1494877171.

[70] S. Jin, X. Zhu, C. Wang, and M. Tomizuka. “Contact Pose Identification for Peg-in-Hole Assembly under Uncertainties”. In: *2021 American Control Conference (ACC)*. 2021, pp. 48–53.

[71] John J. Craig. *Introduction to Robotics: Mechanics and Control*. Pearson Education Inc., 2005. ISBN: 0-13-123629-6.

[72] Jon Owen. *STEP: An Introduction*. Information Geometers Ltd., Winchester, UK, 1997.

[73] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decre, Ruben Smits, Erwin Aertbelien, Kasper Claes, Herman Bruyninckx. “Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty”. In: *International Journal of Robotic Research* 26.5 (2007), pp. 433–455.

[74] Julian Backhaus, Gunther Reinhart. “Digital description of products, processes and resources for task-oriented programming of assembly systems”. In: *Journal of Intelligent Manufacturing* 28 (2017), pp. 1787–1800.

[75] Kazuo Yamafuji. “Development of SCARA Robots”. In: *Journal of Robotics and Mechatronics* 31.1 (2019), pp. 10–15.

[76] Kilian Kleeberger, Jonathan Schnitzler, Muhammad Usman Khalid, Richard Bormann, Werner Kraus, and Marcus F. Huber. “Precise Object Placement with Pose Distance Estimations for Different Objects and Grippers”. In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2021, pp. 4639–4646.

[77] Thomas Kramer and Xun Xu. “STEP in a Nutshell”. In: *Advanced Design and Manufacturing Based on STEP.* London: Springer London, 2009, pp. 1–22. ISBN: 978-1-84882-739-4.

[78] L. I. Lieberman, Michael A. Wesley. “AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly”. In: *IBM Journal of Research and Development* 21.4 (1977), pp. 321–333.

[79] Chiara Lanni and Marco Ceccarelli. “An Optimization Problem Algorithm for Kinematic Design of Mechanisms for Two-Finger Grippers”. In: *The Open Mechanical Engineering Journal* 3 (2009), pp. 49–62.

[80] Remi K. S. Lanza. *Improving and implementing the STEP ISO 10303 standard for design, analysis and structural test data correlation.* PhD Thesis. Norwegian University of Science and Technology (NTNU), Trondheim, 2020.

[81] H. Lee, S. Park, K. Jang, S. Kim, and J. Park. “Contact State Estimation for Peg-in-Hole Assembly Using Gaussian Mixture Model”. In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 3349–3356.

[82] Robert R. Lipman and James J. Filliben. “Testing Implementations of Geometric Dimensioning and Tolerancing in CAD Software”. In: *Computer-Aided Design and Applications* 17.6 (2020), pp. 1241–1265.

[83] Mansour Rahimi, Waldemar Karwowski. “A RESEARCH PARADIGM IN HUMAN-ROBOT INTERACTION”. In: *International Journal of Industrial Ergonomics* 5 (1990), pp. 59–71.

[84] Mathias H. Arbo, Yudha Pane, Erwin Aertbeliën, Wilm Decré. “A System Architecture for Constraint-Based Robotic Assembly with CAD Information”. In: *IEEE International Conference on Automation Science and Engineering* (2018), pp. 690–696.

[85] Maximilian Schmidt. *Konzeption und Einsatzplanung flexibel automatisierter Montagesysteme.* Springer-Verlag, 1992. ISBN: 978-3-540-55025-9.

[86] Andrew T. Miller, Steffen Knoop, Henrik I. Christensen, and Peter K. Allen. “Automatic Grasp Planning Using Shape Primitives”. In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation.* Vol. 2. 2003, pp. 1824–1829.

[87] Model-Based Engineering Forum. *Model-Based Engineering Visual Glossary*. URL: https://modelbasedengineering.com/glossary/. (accessed: 30.05.2022).

[88] Gareth J. Monkman, Stefan Hesse, Ralf Steinmann, and Henrik Schunk. *Robot Grippers*. WILEY-VCH Verlag GmbH & Co. KGaA, 2007. ISBN: 0-408-03561-7.

[89] Morgan S Ohwovoriole, Bernard Roth. “An Extension of Screw Theory”. In: *Journal of Mechanical Design* 103.4 (1981), pp. 725–735.

[90] Nathan Larkin, Andrew Short, Zengxi Pan, Stephen van Duin. “Automated Programming for Robotic Welding”. In: *Transactions on Intelligent Welding Manufacturing* (2020), pp. 48–59.

[91] Nathan W. Hartman, Jesse Zahner. “Extending and Evaluating the Model-based Product Definition”. In: *Grant/Contract Reports (NISTGCR)* (2017).

[92] Nikhil Somani, Markus Rickert, Alois Knoll. “An Exact Solver for Geometric Constraints With Inequalities”. In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 1148–1155.

[93] A. E. Owen. *Flexible Assembly Systems*. Springer Science+Business Media New York, 1984. ISBN: 978-1-4899-0495-9.

[94] Yudha Prawira Pane. *Composing Concurrent Reactive Robot Skills - Integration of Task Specification and Control with Symbolic Planning and CAD Information*. PhD Thesis. KU, Leuven, 2022.

[95] F. Proctor, M. Franaszek, and J. Michaloski. “Tolerances and Uncertainty in Robotic Systems”. In: vol. 2: Advanced Manufacturing. ASME International Mechanical Engineering Congress and Exposition. Nov. 2017. DOI: 10.1115/IMECE2017-70404.

[96] F. Proctor, G. van der Hoorn, and R. Lipman. “Automating robot planning using product and manufacturing information”. In: *Procedia CIRP* 43 (2016), pp. 208–213.

[97] Robin J. Popplestone. “The Edinburgh Designer System as a Framework for Robotics: The Design of Behavior”. In: *Artificial Intelligence for Engineering, Design, Analysis and Manufacturing* 1.1 (1987), pp. 25–36.

[98] Robin J. Popplestone, A. P. Ambler, I. Bellos. “RAPT: A language for describing assemblies”. In: *Industrial Robot: An International Journal* 5.3 (1978), pp. 131–137.

[99] Douglas T. Ross. “Origins of the APT Language for Automatically Programmed Tools”. In: *SIGPLAN Notices* 13.8 (1978), pp. 61–99.

[100] Ruben Smits, Tinne De Laet, Kasper Claes, Herman Bruyninckx, Joris De Schutter. “iTASC: a Tool for Multi-Sensor Integration in Robot Manipulation”. In: *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems* (2008), pp. 426–433.

[101] Sameer Mittal, Muztoba A. Khan, David Romero, Thorsten Wuest. “A Critical Review of Smart Manufacturing & Industry 4.0 Maturity Models: Implications for Small and Medium-sized Enterprises (SMEs)”. In: *Journal of Manufacturing Systems* 49 (2018), pp. 194–214.

[102] Johannes Schmalz, Lucas Giering, Matthias Holzle, Niklas Huber, and Gunther Reinhart. “Method for the Automated Dimensioning of Gripper Systems”. In: *Procedia CIRP* 44 (2016), pp. 239–244.

[103] Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. “Leveraging Model Based Definition and STEP AP242 in Task Specification for Robotic Assembly”. In: *Procedia CIRP* 97 (2021), pp. 92–97.

[104] Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. “Using Semantic Geometric Dimensioning and Tolerancing (GD&T) Information from STEP AP242 Neutral Exchange Files for Robotic Applications”. In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* (2023).

[105] Sharon J. Kemmerer. *STEP: The Grand Experience.* Special Publication (NIST SP). National Institute of Standards and Technology, Gaithersburg, MD, 1999.

[106] Shawn P. Ruemler, Kyle E. Zimmerman, Nathan W. Hartman, Thomas, Jr. Hedberg, Allison B. Feeny. “Promoting Model-Based Definition to Establish a Complete Product Definition”. In: *Journal of Manufacturing Science and Engineering* 139.5 (2016).

[107] K. B. Shimoga. “Robot Grasp Synthesis Algorithms: A Survey”. In: *The International Journal of Robotics Research* 15 (1996), pp. 230–266.

[108] Shimon Y. Nof, Wilbert E. Wilhelm, Hans-Jürgen Warnecke. *Industrial Assembly.* SPRINGER-SCIENCE+BUSINESS MEDIA, B.V., 1997. ISBN: 978-1-4613-7937-9.

[109] Siemens. “JT File Format Reference”. In: Version 10.5 (2023).

[110] S. Simunovic. “Force Information in Assembly Processes”. In: *5th International Symposium on Industrial Robots* (1975).

[111] Nikhil Somani. *Constraint-based Approaches for Robotic Systems: from Computer Vision to Real-Time Robot Control.* PhD Thesis. Technical University, Munich, 2018.

[112] Joao Pedro Carvalho de Souza, Luis F. Rocha, Paulo Moura Oliveira, A. Paulo Moreira, and Jose Boaventura-Cunha. “Robotic Grasping: From Wrench Space Heuristics to Deep Learning Policies”. In: *Robotics and Computer-Integrated Manufacturing* 71 (Apr. 2021).

[113] STEP AP239 Project Committee. “ISO 10303 (STEP) AP 239 edition 3: Application Protocol For Product Life Cycle Support (PLCS)”. In: *Whitepaper* (2015). URL: http://www.ap239.org/. (accessed: 09.05.2022).

[114] STEP AP242 Project Committee. “Development of STEP AP 242 ed2: “Managed Model Based 3D Engineering””. In: *Whitepaper* (2014).

[115] Stephen G. Kaufman, Randall H. Wilson, Rondall E. Jones, Terri L. Calton. “The Archimedes 2 Mechanical Assembly Planning System”. In: *Proceedings of IEEE International Conference on Robotics and Automation* 4 (1996), pp. 3361–3368.

[116] D. R. Strip. “Primitives for Robotic Mechanical Assembly: Force Directed Insertions”. In: *Robotics & Computer-Integrated Manufacturing* 6.4 (1989), pp. 283–286.

[117] Syed M. Abbas, Syed Hassan, Jongwon Yun. “Augmented Reality Based Teaching Pendant for Industrial Robot”. In: *2012 12th International Conference on Control, Automation and Systems*. 2012, pp. 2210–2213.

[118] J. Takahashi, T. Fukukawa, and T. Fukuda. “Passive Alignment Principle for Robotic Assembly Between a Ring and Shaft With Extremely Narrow Clearance”. In: *IEEE/ASME Transactions on Mechatronics* 21.1 (2016), pp. 196–204.

[119] Terri L. Calton. *Complex Assemblies: Rapid Automated Generation of Correct Assembly Sequences*. URL: https://www.automate.org/case-studies/complex-assemblies-rapid-automated-generation-of-correct-assembly-sequences. (accessed: 22.05.2022).

[120] The HDF Group. *The HDF5 Library & File Format*. URL: https://www.hdfgroup.org/solutions/hdf5/. (accessed: 09.05.2022).

[121] The Object Management Group. *SysML*. URL: https://www.omgsysml.org/index.htm. (accessed: 09.05.2022).

[122] The Object Management Group. *XML Metadata Interchange*. URL: https://www.omg.org/spec/XMI/2.5.1/About-XMI/. (accessed: 09.05.2022).

[123] Thomas D. Hedberg, Joshua Lubell, Lyle Fischer, Larry Maggiano, Allison B. Feeney. “Testing the Digital Thread in Support of Model-Based Manufacturing and Inspection”. In: *ASME Journal of Computing and Information Science in Engineering* 16.2 (2016).

[124] Tuan Anh Tran, Eirik Bjørndal Njåstad, Ole Terje Midling, Morten Bjelland, Andrei Lobov. "Generation of Rule-Adhering Robot Programs for Aluminium Welding Automatically from CAD". In: *The International Journal of Advanced Manufacturing Technology* (Accepted).

[125] Ulrike Thomas, Friedrich M. Wahl. "A System for Automatic Planning, Evaluation and Execution of Assembly Sequences for Industrial Robots". In: *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems* 3 (2001), pp. 1458–1464.

[126] Ulrike Thomas, Friedrich M. Wahl. "Assembly Planning and Task Planning—Two Prerequisites for Automated Robot Programming". In: *Robotic Systems for Handling and Assembly*. Springer-Verlag Berlin Heidelberg, 2010, pp. 333–354. ISBN: 978-3-642-16784-3.

[127] Weidong Shen, Tianliang Hu, Chengrui Zhang, Yingxin Ye, Zhengyu Li. "A Welding task data model for intelligent process planning of robotic welding". In: *Robotics and Computer Integrated Manufacturing* 64 (2020), pp. 1–12.

[128] Weidong Shen, Tianliang Hu, Chengrui Zhang, Yingxin Ye, Zhengyu Li. "A welding task data model for intelligent process planning of robotic welding". In: *Robotics and Computer Integrated Manufacturing* 64 (2018).

[129] D. E. Whitney. "Quasi-Static Assembly of Compliantly Supported Rigid Parts". In: *Journal of Dynamic Systems, Measurement, and Control* 104.1 (1982), pp. 65–77.

[130] William R. Tanner. "Chapter 29: Product Design and Production Planning". In: *Handbook of Industrial Robotics*. John Wiley & Sons, Inc., 1999, pp. 529–542. ISBN: 978-0-471-17783-8.

[131] Yudha Pane, Mathias H. Arbo, Erwin Aertbeliën, Wilm Decré. "A System Architecture for Constraint-Based Robotic Assembly with Sensor-Based Skills". In: *IEEE Transactions on Automation Science and Engineering* 17.3 (2020), pp. 1237–1249.

[132] Zhongxiang Zhou, Rong Xiong, Yue Wang, Jiafan Zhang. "Advanced Robot Programming: a Review". In: *Current Robotics Reports* 1 (2020), pp. 251–258.

Appendix A.

Publications

A.1. Paper I:

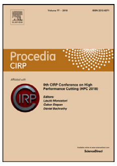
Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. (2021) “Leveraging Model Based Definition and STEP AP242 in Task Specification for Robotic Assembly,” Procedia CIRP. vol. 97, pp. 92–97.

Procedia CIRP 97 (2020) 92–97

Contents lists available at ScienceDirect

Procedia CIRP

journal homepage: www.elsevier.com/locate/procir



Leveraging model based definition and STEP AP242 in task specification for robotic assembly

Shafi K. Mohammed[a,*], Mathias H. Arbo[b], Lars Tingelstad[a]

[a] Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology (NTNU), Trondheim 7491, Norway
[b] Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim 7491, Norway

ARTICLE INFO

Article history:
Received 28 September 2019
Revised 24 April 2020
Accepted 19 May 2020

Keywords:
Robotics
Assembly
Constraint-based robot programming
Model based definition
Product manufacturing information
STEP AP242
ISO 10303

ABSTRACT

This article explores using aspects of STEP AP242 for constraint-based robot programming for assembly operations. Industry 4.0 envisions smart, connected factories where all the operations are connected by an unbroken thread of product and process data. As part of these efforts, many industries are adopting Model Based Definition and STEP AP242. STEP AP242 is an exchange format that allows Model Based Definition where Product Manufacturing Information is directly associated with the 3D CAD model. This article relates the geometric and assembly constraints from the CAD model to motion constraints on the robot during the assembly process. This article also discusses the use of Product Manufacturing Information from STEP AP242 files for automatic robot programming. The results are showcased with a prototype for a motor assembly scenario.

1. Introduction

The new industrial revolution, Industry 4.0, brings aspects of information technology into the manufacturing industries, with high product customization, information gathering cyber-physical systems, and a short time to deploy. The success of Industry 4.0 depends on the integration of design and manufacturing operations.

One of the key enablers for Industry 4.0 is the availability of product data to various stakeholders involved in the product life cycle process. The completeness of the available data affects the efficiency of product manufacturing operations both in terms of time and product quality.

A large amount of data is created in the initial phases of product development and design. The designers use CAD software to design a product and create 3D models. 2D manufacturing drawings are created from the 3D CAD models, and product manufacturing information (PMI) is added to the drawings. These manufacturing drawings are shared with the downstream operations, resulting in many communication gaps between the designer and the manufacturer that give rise to delays and quality issues.

Many downstream operations like inspection need manual intervention to study the 2D drawings, extract the necessary data, and recreate it to match their needs. The recreation of information is time-consuming and error-prone due to human involvement. In the scenario of Industry 4.0, the manual interpretation and recreation of data should be avoided as much as possible. The solution for this is reusing the data created in the product development stage in downstream manufacturing operations without recreation.

The recreation of information can be eliminated by employing model-based definition (MBD) in which PMI is directly appended to the 3D CAD model during the design phase. The critical dimensions, geometric dimensioning and tolerancing (GD&T), surface finish, and other needed information, are semantically added to the features of the 3D CAD model. This information is then available for direct use in downstream operations like NC machining and inspection Herron (2013); Uros Urbas, Rok Vrabic, Nikola Vukasinovic (2019).

Traditionally there are three methods of programming robots: (1) teach-pendant programming, (2) offline programming, and (3) programming by demonstration. A limitation of these approaches is that they cannot be used for the automated programming of robot operations – the robots can be used to perform only pre-programmed tasks. They cannot be used to control the manipulators to perform in new/unexpected scenarios. This severely limits the use of robots in small and medium enterprises that have a wide variety of products.

Future smart factories need intelligent robotic systems that can be used to complete various operations working in mixed environ-

[*] Corresponding author.
E-mail address: shafi.k.mohammed@ntnu.no (S.K. Mohammed).

https://doi.org/10.1016/j.procir.2020.05.209
2212-8271/

ments with humans and other machines. One approach to solving this and overcome the limitations of traditional robot programming methods is constraint-based robot programming. In this approach, the robotic task is specified in terms of relations between components in the assembly, as well as with the environment, and not directly by point-to-point motion or other motion primitives. However, this requires that the robotic system is able to detect and extract the constraints involved in the task Somani (2018).

Integrating CAD in robot programming systems is a longstanding problem in robotics. Early systems capable of extracting relevant geometric information includes the Autopass, Archimedes 2, and HigLap frameworks.

Autopass is the earliest robot programming language based on CAD and focuses on the product being assembled, the tools, fixtures and the assembly tasks. An assembly world model is created from the geometric data and updated at each step of the assembly operation. The users plan the assembly sequence and specify the assembly operations as if they are preparing an assembly instruction manual without bothering about the actual motion of the manipulator Lieberman and Wesley (1977).

Archimedes 2 is an assembly planning system that uses 3D CAD data to facilitate assembly planning and robot control. The 3D data from CAD files is used as an input to planning and simulation operations. The user can plan the assembly operations with the help of a planner and an illustrator, which simulates the operations. A translator converts these high-level assembly plans into control code for robotic assembly Kaufman, Wilson, Jones, Calton (1996).

HighLap is an assembly planning system that automatically generates the assembly plans from the CAD data. It uses an assembly by disassembly method to decompose the assembly tasks into primitive robot skills Mosemann, Friedrich M. Wahl (2001); Thomas, Wahl (2001).

SMERobotics is an EU funded consortium aimed at enabling robotics for small to medium sized enterprises. This consortium presents a method for rapid robot programming based on mating constraints that are not extracted from the STEP or CAD file, but rather specified by a shop floor worker Perzylo et al. (2019). The part-oriented programming method showed good time-saving when compared to classical programming methods Perzylo et al. (2016). The system can plan assembly paths in the motion null-space formed by the mating constraints Somani (2018); Somani et al. (2017).

More recent frameworks for constraint-based robot programming are the iTaSC Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decre, Ruben Smits, Erwin Aertbelien, Kasper Claes, Herman Bruyninckx (2007) and eTaSL/eTC Erwin Aertbelien, Joris De Schutter (2014).

The iTaSC framework uses feature coordinates to define object constraints and uncertainty coordinates to estimate geometric uncertainties. This achieves instantaneous task specification for complex robotic applications with sensor interaction and geometric uncertainties Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decre, Ruben Smits, Erwin Aertbelien, Kasper Claes, Herman Bruyninckx (2007); Smits, De Laet, Kasper Claes, Herman Bruyninckx, Joris De Schutter (2008). eTaSL is a task specification language, and eTC is a corresponding controller both based on expression graphs. Expression graphs are a tree-like data structure that specifies the geometric relations between objects and supports the computation of the Jacobians using automatic differentiation Erwin Aertbelien, Joris De Schutter (2014).

In this paper, we explore the use of model-based definition and the use of the STEP AP242 exchange format for constraint-based robot programming for assembly operations. The product design information in the form of 3D CAD models (both the part and assembly) carries rich and useful information about the components and their constraints. The annotated models with the PMI enriches the geometric data of the 3D CAD and imparts meaning to these constraints. This data can be readily extracted and used to automate the programming and control of robots saving time and cost. The presented approach employs the eTaSL/eTC framework and is based on exporting the assembly constraints directly from the 3D CAD system and using these to generate the motion constraints of the robotic system automatically. A schematic presentation of our approach is shown in Fig. 1.

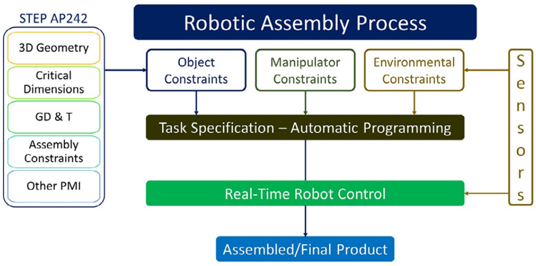


Fig. 1. Constraint-based robot programming using STEP AP242.

The paper is structured as follows. Section 2 presents the STEP AP242 standard and how assembly constraints can be modelled and exported from 3D CAD data and Section 3 presents the eTaSL/eTC framework in more detail and shows how assembly constraints can be used to form motions constraints. A test case is presented in Section 4. A discussion of the presented approach is presented in Section 5. The paper is concluded in Section 6.

2. Extracting assembly constraint information from STEP AP242 exchange files

The product data created during the design stage should be available to all the stakeholders in the downstream processes of the product life cycle phases. The 3D models can be shared with the downstream operations using native CAD files or neutral formats.

The standard for the Exchange of Product Model Data (STEP) is one of the standard neutral file formats used by the industry. It is a set of ISO (International Standards Organization) standards under ISO 10303. The two application protocols (APs) *STEP AP203 – Configuration controlled 3D designs of mechanical parts and assemblies* and *STEP AP214 – Core data for automotive mechanical design processes* are being used by aerospace and defense, and automotive industries, respectively. These two APs have overlapping scopes, and their latest editions started converging towards each other. Due to this, the application protocol *AP242 – Managed model based 3D engineering* was developed. This is a significant AP which combines and replaces AP203 and AP214 (along with few other APs under ISO 10303) Development of STEP AP 242 ed2: (2014); ISO 10303-242:2014 (0000); Kramer and Xu (2009).

STEP AP242 defines the use of semantic PMI for product definition. Under this standard, product data like GD&T, roughness specifications, welding details, direction features, contacting tangents, and hull features are readily available at both the part and assembly level.

As the industry is moving towards smart manufacturing, the adoption of STEP AP242 is increasing. AeroSpace and Defence Industries Association of Europe (ASD) recommends the use of latest editions of this standard for “exchange, long term archiving and transfer to downstream processes of CAD data (mechanical design, including composite) and associated configuration (PDM) data” Jean-Yves Delaunay (2018).

Table 1
Assembly Constraints and applicable geometric entities.

Assembly Constraint Entity	Line	Plane	Cylindrical/Conical/ Spherical Surfaces
Parallel_assembly_constraint	*	*	
Parallel_assembly_constraint_with_dimension	*	*	
Surface_distance_assembly_constraint_with_dimension	*	*	
Angle_assembly_constraint_with_dimension		*	
Perpendicular_assembly_constraint	*	*	
Incidence_assembly_constraint	*		*
Coaxial_assembly_constraint			*
Tangent_assembly_constraint	*	*	*

2.1. Relevant features of STEP AP242 exchange files

The following types of information are available in a STEP AP242 exchange file: Geometric data, assembly constraints, bill of materials, geometric dimensioning and tolerancing (GD&T) details, and other product manufacturing information (PMI) in the form of annotations.

Some of the information like the geometric details of the product are readily available in the STEP file but some information has to be inferred from the available data. For robotic assembly applications, geometric definition of the product alone is not much helpful. Product structure and assembly constraints are needed along with the geometric details.

2.1.1. Assembly constraints

The types of assembly constraints available in STEP AP242 ISO 10303-242:2014 (0000) are:

- *Fixed_constituent_assembly_constraint*: This is applied to the part which is fixed in the assembly and all other parts are placed with respect to this part. The coordinate system and origin of this part coincides with the world-coordinate system and origin of the assembly.
- *Parallel_assembly_constraint*: this is used to define the parallel relationship between the mating features of the parts. This is applicable to lines and planes.
- *Parallel_assembly_constraint_with_dimension*: this is a subtype of parallel_assembly_constraint with a 'distance_value' to define the distance between the parallel features.
- Surface_distance_assembly_constraint_with_dimension: this is used to define the normal distance between lines and planes of the mating parts.
- *Angle_assembly_constraint_with_dimension*: this defines the angle between the line and plane features.
- *Perpendicular_assembly_constraint*: this is used to apply the perpendicularity constraint between the mating lines and planes.
- *Incidence_assembly_constraint*: this is used when a line is incident on a plane or to define the concentricity of circular, cylindrical, conical or spherical features.
- *Coaxial_assembly_constraint*: this constraint is used when the axes of cylindrical or conical features have the same direction.
- *Tangent_assembly_constraint*: this defines the tangency of circular features with lines, planes or other circular features.

Table 1 shows all the assembly constraints available in STEP schema and the geometric features related by these constraints.

The assembly constraints available in the commercial CAD software correspond to these types of constraints. However, even though the STEP AP242 standards supports assembly constraints, when the assembly is exported as a STEP file from a CAD software, e.g., SolidWorks or Siemens NX, the constraints are not explicitly available in the STEP file.

2.1.2. Bill of materials

The bill of materials (BoM) is beneficial information in understanding the product structure. It gives the number of components used in the assembly and how many times each component is used. BoM can be explicitly added to the assembly as a table in the annotations.

2.1.3. GD&T and other PMI

GD&T and other manufacturing information can be added to the part file in the form of annotations, which increases the understanding of the product and helps in creating robust automated robot programming systems. In many assemblies, the mating fits can be inferred from the GD&T, which helps in understanding the force constraints on the robot during the assembly operation.

2.2. Extracting constraint information

The required constraint information is extracted from the STEP file and passed to the eTaSL framework for task specification. The constraint information includes parts in the assembly, the relative position of the parts to each other, assembly constraints, mating features of the parts, and their positions. The steps involved in this process are:

1. Identifying the parts in the assembly: the total number of parts and their instances are identified from the STEP file. This gives the overall product structure. Alternately this can be identified from the BoM if it is added as a table in the annotations.
2. Establishing the global coordinate frame: this is the coordinate frame to which the entire assembly is defined. By default this coordinate frame is defined at the origin.
3. Identifying the fixed constituent part: this is the immovable part in the assembly to which all other parts are placed in the assembly. Generally, the part coordinate frame of the fixed part coincides with the global coordinate frame of the assembly.
4. Get the next part in the assembly: it is assumed that the assembly sequence is given by the order in which the parts are added to the assembly or the order in which the parts appear in the STEP file.
5. Identify the part coordinate frame: this coordinate frame gives the relative position of each part in the assembly. All the part features are defined with respect to this coordinate frame.
6. Find the assembly constraints: identify all the assembly constraints applied on a part.
7. Establish feature coordinate frames: for each assembly constraint, find the corresponding mating features. Establish appropriate coordinate frames for these mating features. For simple surfaces, these can be directly identified from the STEP file. These feature coordinate frames are used in eTaSL to define the robotic tasks.
8. For all the parts in the assembly, repeat steps 4 to 7.

Table 2
Summary of distance measures – eTaSL.

Description	Distance Measure expression
Parallel (direction/normal/axis $\mathbf{d}_1, \mathbf{d}_2$	$\|\|\mathbf{d}_1 \times \mathbf{d}_2\|\|$
Perpendicular (direction/normal/axis $\mathbf{d}_1, \mathbf{d}_2$)	$\|\mathbf{d}_1 \cdot \mathbf{d}_2\|$
Distance point ($\mathbf{p}_1$) – line ($\mathbf{p}_1, \mathbf{d}_2$)	$\|\|\mathbf{p}_1 - \mathbf{p}_2 - ((\mathbf{p}_1 - \mathbf{p}_2)) \cdot \mathbf{d}_2)\mathbf{d}_2\|\|$
Distance line ($\mathbf{p}_1, \mathbf{d}_1$) – line ($\mathbf{p}_2, \mathbf{d}_2$)	$\begin{cases} \text{Distance point } (\mathbf{p}_1)\text{-line}(\mathbf{p_2}, \mathbf{d_2}), & \text{if } \|\|\mathbf{d}_1 \times \mathbf{d}_2 = 0\|\| \\ \frac{\|(\mathbf{p}_1 - \mathbf{p}_2) \cdot (\|\|\mathbf{d}_1 \times \mathbf{d}_2\|\|)\|}{\|\|\mathbf{d_1} \times \mathbf{d}_2\|\|}, & \text{otherwise} \end{cases}$
Distance line ($\mathbf{p}_1, \mathbf{d}_1$) – plane($\mathbf{p}_2$,$\mathbf{n}_2$)	$\begin{cases} \|(\mathbf{p}_1 - \mathbf{p}_2) \cdot \mathbf{d}_1\|, & \text{if } \mathbf{d}_1 \cdot \mathbf{n}_2 = 0 \\ 0, & \text{otherwise} \end{cases}$

3. eTaSL/eTC for assembly constraints

eTaSL Erwin Aertbelien, Joris De Schutter (2014) is a Lua-based task specification language, eTC is a simultaneous hierarchical task controller realization for the task specifications. The controller works by inverting the differential kinematics of the constraints using a quadratic optimization program, which is solved for joint velocity commands that are sent to the robot. In the cases where a joint velocity command is not available, the joint velocities are integrated to achieve joint position commands.

3.1. Task specification

The task specification is a Lua-based eTaSL script. A single eTaSL script defines a set of tasks, robot variables, support for continuous sensor inputs, event triggering when sensor or controller conditions are achieved, and continuous state outputs for monitoring and debugging. eTaSL defines tasks where the task equations are either driven to zero or driven to remain within an upper or lower limit. The set-based tasks enable defining work-cell related tasks such as remaining within a workspace or keeping the robot in an elbow up configuration. For more complex task specifications, eTaSL also supports feature variables that are controller-internal and sensor inputs. This allows for sensor-guided motion, such as compliant assembly strategies Hauan Arbo, Pane, Aertbeliën, Decré (2018), and sensor-based event triggering. Relating CAD assembly constraints to force-based robot tasks is an ongoing research topic Pane, Hauan Arbo, Aertbeliën, Decré (0000), which currently goes through a selection process of available assembly skills rather than using the assembly constraint directly.

3.2. Distance measures

The assembly constraints are defined between geometric elements whose placement is defined relative to the part coordinate frame. The parts either have predefined locations in the work cell, are located with vision-based localization systems, or are rigidly grasped by the robot. The relative placement and parameters of the geometric elements are extracted from the STEP file, and used to describe the assembly relative to the robot gripper in the task specification.

Establishing a constraint between parts is considered to be the same as minimizing a distance measure between the geometric elements on the parts. This is achieved by selecting a joint velocity on the robot which imposes an exponential convergence of the distance measure. For joint variables $\mathbf{q} \in \mathbb{R}^n$,and scalar distance function e,this is achieved by ensuring

$$\dot{e}(t, \mathbf{q}) = \frac{\partial e}{\partial \mathbf{q}} \dot{\mathbf{q}} = -Ke(t, \mathbf{q}) - \frac{\partial e}{\partial t}$$

holds for the choice of $\dot{\mathbf{q}}$that will command the robot. $K$is a tunable gain that defines the rate of convergence to the task. Slack variables can be included to handle incongruous tasks.

A sample of the distance measures used to implement the constraints described in Table 1 in eTaSL are given in Table 2. With $\mathbf{p} \in \mathbb{R}^3$being a point relative to the shape representation, and $\mathbf{n} \in \mathbb{R}^3$being a vector in the reference frame of the shape representation, lines are represented by $(\mathbf{p}, \mathbf{d})$describing a point $\mathbf{p}$on the line and its direction $\mathbf{d}$. These are implemented as eTaSL expressions using the `Vector`datatype, which is normalized to unit vectors for directions, and geometric entities implemented in eTaSL follow the parametrization described in the geometry schema ISO 10303-42.

4. Test case

In this test case a rotor is inserted into a motor housing as part of a motor. The motor assembly is shown in Fig. 2. The test case is implemented using the *etasl_ros_control* library Tingelstad, Hauan Arbo (2020).

The purpose of this test case is to show the generation of robot motion derived directly from STEP AP242 assembly constraints translated into eTaSL tasks, using the above-mentioned distance measures, for programming robotic assembly. The rotor has an *Incidence_assembly_constraint* between its cylinder axis and the cylinder axis of the hole in the housing, and the bottom plane of the rotor has an *Incidence_assembly_constraint* with the bottom plane it is touching. The assembly sequence is shown in Fig. 3. The part is grasped and lifted above the housing in frames 1 and 2; the assembly constraints are then used to guide the motion during frames 3–6.

The norm of the common normal between the axes, the distance between the axes, and the distance between the planes during assembly are given in Fig. 4. The incidence constraint between the planes also imposes the two plane normals to be parallel, but that is not included as it has the same behavior as the norm of the common normal between the cylinder axes.

5. Discussion

By using the STEP AP242 directly as tasks in the reactive eTaSL controller, rather than indirectly informing the choice of assembly skill Pane, Hauan Arbo, Aertbeliën, Decré (0000), or to define spaces to plan within Somani (2018), specific issues arise.

The task convergence formulation of eTaSL is an exponential convergence; this means that one must tune the task specification such that the tasks that define the alignment process, e.g., the line-line incidence of the axes in the rotor example, have converged before the tasks that ensure the mating and surface contacts, e.g., the plane-plane incidence constraint. The exponential convergence formulation will also incur a high velocity at the beginning and a low velocity at the end of the process. The high initial velocity, and discontinuity in acceleration, might present some problems with physical equipment.

These issues suggest that a more advanced controller formulation than the simple exponential convergence may be required, or that a planning process must be used to figure out which constraints must be achieved before others.

As mentioned earlier, the force requirements for the assembly tasks can be estimated from the mating fits based on the GD&T information. Other process-specific requirements and constraints can

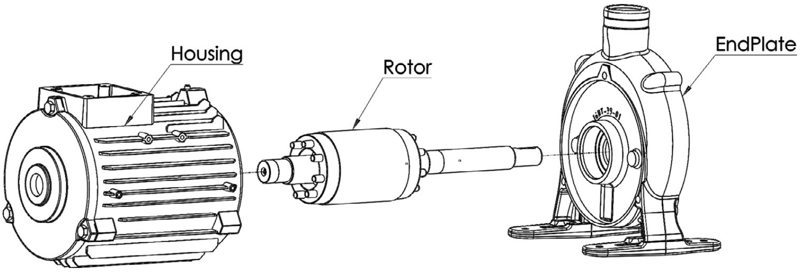


Fig. 2. Motor assembly – exploded view.

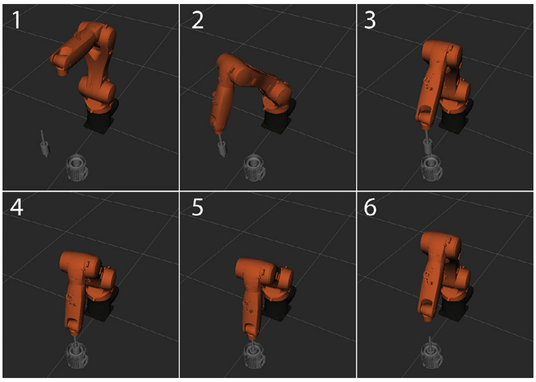


Fig. 3. Assembly sequence, frames 1 and 2 show the part being grasped and lifted above the housing, frames 3–6 show the motion being guided by the assembly constraints for inserting the rotor.

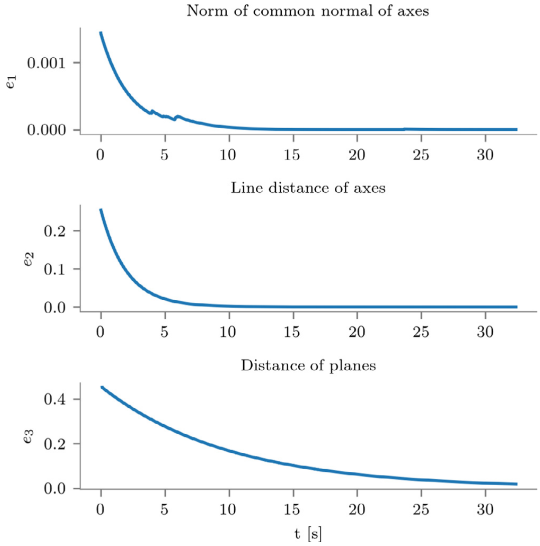


Fig. 4. The distance measures over time: the common normal of the rotor's cylinder axis and the housing's hole cylinder axis, the line distance of the axes, and the distance between the bottom plane of the rotor and the bottom plane of the housing.

be added to the STEP file in the form of annotations and can be used in the task specification. The annotations can be added as per the ontological approaches to improve data reuse in downstream operations.

The second edition of STEP AP242 will add more PMI features for welding, GD&T of threads, complex holes, and other enhancements for new manufacturing features Development of STEP AP 242 ed2: (2014). These new enhancements, along with other features like kinematics, can be explored, and methods can be developed to use them for effective automation of manufacturing and assembly.

6. Conclusions and future research

In this paper, we have presented how product information can be extracted from STEP AP242 exchange files and used in task specification for robotic assembly in general, and a motor assembly task in particular. This is the first step towards leveraging model-based definition and STEP AP242 in automatic programming of robotic assembly tasks.

The motor assembly test case was implemented using eTaSL/eTC – a framework for constraint-based robot programming – where the assembly constraints derived from the exchange file were converted to distance measures in the task specification. That is, the assembly constraints of the 3D CAD model of the motor were converted to motion constraints of the robot.

One issue that arose in the presented approach was due to the exponential convergence formulation: The task has a high initial velocity and low final velocity and thus slow convergence when the distance measures, or task errors, are low. This high initial velocity might present some problems with physical equipment, while the slow convergence is unfortunate due to possibly high cycle times. These issues will be addressed in future research.

Acknowledgments

The work reported in this paper is partially based on activities within centre for research based innovation SFI Manufacturing, and partially funded by the Research Council of Norway under contract number 237900.

References

Aertbelien, R., Schutter, J.D., 2014. eTaSL/eTC: a constraint-based task specification language and robot controller using expression graphs. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1540–1546.
Development of STEP AP 242 ed2:, 2014. Managed model based 3D engineering.
Delaunay, J., 2018. Use of ISO 10303 AP242 in the European A&D industries. Global Product Data Interoperability Summit 2018.
Hauan Arbo, M., Pane, Y., Aertbeliën, E., Decré, W., 2018. A system architecture for constraint-based robotic assembly with CAD information. In: Proceedings of the IEEE International Conference on Automation Science and Engineering, pp. 690–696.

Herron, J.B., 2013. Re-Use Your CAD: The Model-Based CAD Handbook. CreateSpace Independent Publishing Platform.

ISO 10303-242:2014, Industrial automation systems and integration “Product data representation and exchange – Part 242: application protocol: managed model-based 3D engineering”. Int. Org. Standard.

Kaufman, S.G., Wilson, R.H., Jones, R.E., Calton, T.L., 1996. The archimedes 2 mechanical assembly planning system. In: Proceedings of IEEE International Conference on Robotics and Automation, 4, pp. 3361–3368.

Kramer, T., Xu, X., 2009. STEP in a Nutshell. Springer London, London, pp. 1–22. doi:10.1007/978-1-84882-739-4_1.

Lieberman, L.I., Wesley, M.A., 1977. AUTOPASS: an automatic programming system for computer controlled mechanical assembly. IBM J. Res. Dev. 21 (4), 321–333.

Mosemann, H., Friedrich, M.W., 2001. Automatic decomposition of planned assembly sequences into skill primitives. IEEE Trans. Robot. Autom. 17 (5), 709–718.

Pane, Y. Hauan Arbo, M. Aertbeliën, E. Decré, W., A system architecture for constraint-based robotic assembly with sensor-based skills. IEEE Trans. Autom. Sci. Eng..

Perzylo, A., Rickert, M., Kahl, B., Somani, N., Lehmann, C., Kuss, A., Profanter, S., Beck, A.B., Haage, M., Rath Hansen, M., Nibe, M.T., Roa, M.A., Sornmo, O., Gestegard Robertz, S., Thomas, U., Veiga, G., Topp, E.A., Kessler, I., Danzer, M., 2019. Smerobotics: smart robots for flexible manufacturing. IEEE Robot. Autom. Mag. 26 (1), 78–90.

Perzylo, A., Somani, N., Profanter, S., Kessler, I., Rickert, M., Knoll, A., 2016. Intuitive instruction of industrial robots: semantic process descriptions for small lot production. In: Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2293–2300.

Schutter, J.D., Tinne, D.L., Rutgeerts, J.o, Decre, W., Smits, R., Aertbelien, E., Claes, K., Bruyninckx, H., 2007. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. Int. J. Robot. Res. 26 (5), 433–455.

Smits, R., De Laet, T., Claes, K., Bruyninckx, H., De Schutter, J., 2008. iTASC: a tool for multi-sensor integration in robot manipulation. In: Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 426–433.

Somani, N., 2018. Constraint-based approaches for robotic systems: from computer vision to real-time robot control. Technical University, Munich.

Somani, N., Rickert, M., Knoll, A., 2017. An exact solver for geometric constraints with inequalities. IEEE Robot. Autom. Lett. 2 (2), 1148–1155.

Tingelstad, L., Hauan Arbo, M., 2020. etasl_ros_control doi:10.5281/zenodo.3610916.

Thomas, U., Wahl, F.M., 2001. A system for automatic planning, evaluation and execution of assembly sequences for industrial robots 3, 1458–1464.

Urbas, U., Vrabic, R., Vukasinovic, N., 2019. Displaying product manufacturing information in augmented reality for inspection. Procedia CIRP 81, 832–837.

A.2. Paper II:

Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. (2021) “Constraint Identification from STEP AP242 files for Automated Robotic Welding,” 2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), 2021, pp. 277-282.

The preprint version of the paper is included here.

Constraint Identification from STEP AP242 files for Automated Robotic Welding*

Shafi K. Mohammed
Department of Mechanical and Industrial Engineering
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
shafi.k.mohammed@ntnu.no

Mathias H. Arbo
Department of Engineering Cybernetics
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
mathias.arbo@ntnu.no

Lars Tingelstad
Department of Mechanical and Industrial Engineering
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
lars.tingelstad@ntnu.no

***Abstract*—This article proposes the use of STEP AP242 for identifying and extracting the relevant product data for constraint-based programming of welding robots. Most of the product data is created during the design phase of its life cycle. This data can be reused in downstream processes employing Digital Thread as part of Industry 4.0 practices. This article proposes using STEP AP242 Edition 2 neutral file exchange format for extracting geometry and Product Manufacturing Information. This article discusses various types of constraints and welding information that can be included in STEP AP242 files and extracted for automated programming and real-time control of robotic welding operations.**

Keywords—robotics, welding, constraint-based robot programming, model based definition, welding annotations, STEP AP242, ISO 10303

I. INTRODUCTION

One of the earliest and most successful adoptions of robots in manufacturing is for welding tasks. However, this is limited to large-scale industries like automotive, where the welding task is repetitive. In the case of small and medium industries where product variation is high, robotic welding is limited. This is due to the time taken to change the robot programs for changing parts and that it requires highly skilled programmers [1]. By adopting good design practices and better part management [2], small and medium industries can make the parts simple enough to adopt robot welding. As part of Industry 4.0, the manufacturing industry is moving towards complete automation. It is highly recommended that the industry adoption of robot welding increases in line with the automation in other manufacturing operations.

If the upstream operations are using CNC and other automated processes, then the robot welding downstream will be easy and can capitalize on the advantages of upstream processes [3]. Another aspect of Industry 4.0 is establishing a connected Digital Thread (DT) where an unbroken information and material flow is established throughout the extended enterprise. This tries to avoid the recreation of information at each stage of a product life cycle by reusing the product information created at earlier stages. Shafi et al. [4] presented a method in which product information from the design phase is used for automated robotic assembly. This paper aims at reusing the welding information from the product design phase for robotic welding. The methodology used for this purpose is presented in Fig. 1.

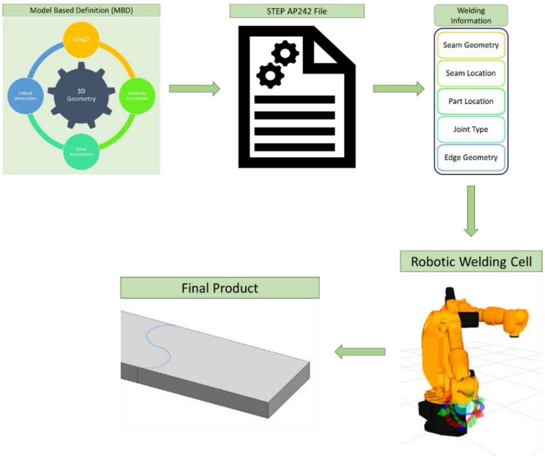


Fig. 1. Automated robotic welding using Information from STEP AP242 Files

This paper presents a method to include the welding annotations as Unicode string in the STEP AP242 files and how it can be reused. This paper discusses how the entities defined in the latest edition of STEP AP242 can be used for this purpose. A method to find the seam geometry and location is also discussed. It also explains how this information can be reused for constraint definition for robot control.

This paper is structured as follows. Section 2 presents a brief review of constraint-based robot programming systems and programming approaches for welding robots. Section 3 describes the Model Based Definition (MBD) methodology, and Section 4 discusses the information available in STEPAP242 files that can be extracted. The following Section 5 presents how

*This work was supported by the Norwegian Research Council infrastructure project MANULAB: Norwegian Manufacturing Research Laboratory under grant 269898. And partially funded by the Research Council of Norway under contract number 237900.

welding information can be included in STEPAP242 for robotic welding and how to reuse it. Section 6 briefly describes the use of the welding information for constraint definition. Section 7 presents the implementation method. The concluding remarks are in Section 8.

II. Robotic Programming

Classical robot programming defines the motion and feed-back control of the robot with regards to either the joint coordinates or some defined task or world coordinates. This requires dedicated scripts or functions for executing motions and keeps the robot motion separate from the part's design. The resulting program is easier to optimize for large volume productions but is not flexible regarding variations in the design of the part. The limitations mentioned above can be overcome by constraint-based programming. In this approach, the constraint information can be extracted from the CAD files resulting in a tighter coupling of design and production. Some of the frameworks which integrate CAD with constraint-based robot programming are Autopass [5], Archimedes2 [6], HighLap [7], iTaSC [8] and eTaSL/eTC [9].

Norberto Pires et al. [10], [1] presented a CAD interface for programming welding robots, where 3D CAD models of the workbench and the product are used to extract the geometric details and relative position of the product in the work environment. Other necessary information like the trajectories to be followed by the manipulator to complete the weld and welding parameters are added by a user. All this information is stored in a DXF file and used to generate the robot program. Shafi et al. [4] extracted constraint information from 3D CAD (STEP) files for task definition in eTaSL/eTC for automated robot programming for assembly operations. Larkin et al. [11] described an automated offline programming method for programming welding robots using CAD models. In this approach, the weld paths and other process parameters are added to the 3D CAD model separately before exporting it to a neutral format. Then this information from these neutral CAD files is extracted to generate offline programs for robotic welding. Weidong et al. [12] developed a welding data model based on the STEP standard using EXPRESS data modeling language. An EXPRESS schema 'Weld_task_model' is used to capture the relevant welding information, including material, weld geometry, and quality. These STEP files are used for offline programming and task planning of welding operations.

III. Model Based Definition (MBD)

Traditionally, designers prepare 2D manufacturing drawings that are shared with the downstream operations. The necessary information is extracted and recreated manually from the manufacturing drawings. This manual intervention at the downstream operations can be avoided by adopting Model Based Definition (MBD) as part of a Digital Thread implementation. In MBD, the Product Manufacturing Information (PMI) like critical dimensions, geometric dimensioning and tolerancing (GD&T), welding annotations, surface finish, and other needed information, are semantically added to the relevant features of the 3D CAD model [13]. The semantic attachment of the annotations to 3D CAD models increases the clarity and reduces confusion as in the understanding of 'arrow side' and 'other side' compared to a 2D manufacturing drawing. This PMI can be directly reused in many downstream processes. Depending on the ease with which the parameters can be controlled to achieve a good quality weld, Norberto Pires et al. [1] classified them as follows:

- Primary inputs: These variables can be controlled during the welding process to achieve a good quality weld. Depending on the welding process, these can be voltage or welding speed.
- Secondary inputs: These parameters are fixed once the welding process is selected. In the case of Gas Tungsten Arc Welding (GTAW), the secondary inputs will be shielding gas or filler material.
- Fixed inputs: These are the fixed parameters that cannot be changed by the user. These depend on the product(geometry, material, etc.) and welding process selected.

Some of this information is already present in the 3D CAD model of the product like weld geometry. Most of the process parameters can be included in the MBD of the product as annotations which can later be used in the automatic programming of the welding robots.

IV. Identification and Extraction of Welding Constraints From STEP AP242 Files

The standard for the Exchange of Product Model Data (STEP) is a standard neutral file format that enables the sharing of product information among various stakeholders. It is the ISO (International Standards Organization) standards 10303: Automation systems and integration – Product data representation and exchange. The application protocolAP242: Managed model-based 3D engineering deals with 3D semantic PMI. It replaced the two earlier application protocols STEP AP203 – Configuration controlled 3D designs of mechanical parts and assemblies [14] and STEP AP214 – Core data for automotive mechanical design processes [15]. The second edition of this application protocol (AP) was released in early 2020 [16]. The PMI can be added to the 3D CAD model, as suggested in ASME Y14.41–2019: Digital Product Definition Data Practices [17]. For constraint-based robot programming, the presentation aspects of PMI are not relevant and only semantic elements are enough to extract the needed information. The following STEP AP242 features can be used to extract the required welding information.

- Geometric Details: The geometric product details are readily available in the STEP file and can be used directly for identifying the various features and edges and their parametrization.
- Assembly Information: Product structure, relative positions of parts in the assembly, type of mating (like bolted joint, riveting, and welding) can be extracted from STEP AP242 files. The extraction of assembly information and its application in constraint-based programming for assembly tasks is presented by Shafi et al. [4].
- Critical Dimensions and Geometric Dimensioning and Tolerancing (GD&T): Critical part dimensions and

GD&T are added to the 3D CAD model in the form of annotations. These can be used to extract the required information like part thickness and length, which can be used for defining the welding parameters.

- Welding Annotations: The second edition (Ed2) of STEP AP242 supports welding annotations. STEPAP242 supports adding the welding information as per AWS A.24 edition 2012 Standard Symbols for Welding, Brazing, and Nondestructive Examination [18] and ISO2553:2013 – Welding and allied processes – Symbolic representation on drawings – Welded joints [19].
- Product Properties: Properties like product material and weight can be included in the STEP file and later used to automate the welding process selection or estimate the forces on manipulators during welding.
- Notes: Other process-related information can be added to the 3D CAD model as text annotation. These can include the voltage requirements or any other critical information that can be used to control the robot during the welding process.

A. *Welding Annotations*

In the second edition of STEP AP242 standards, semantic representation of welding annotations is enabled. Using these new features welding information can be added as per AWS A.24 [18] and ISO 2553:2013 [19]. One issue with the STEP standard is that even the latest edition did not incorporate all the welding parameters needed to complete the welding operation effectively. Due to this limitation, many parameters need to be added separately as user defined entities in a STEP file.

V. Methods To Include Welding Information in the STEP Assembly File

This section presents the ways in which the welding information can be included in the STEP assembly file.

A. *Using Custom EXPRESS Schema*

Required welding information can be included in the STEP assembly files by defining a separate schema for this purpose. This method was adopted by Weidong et al. [12]. Weidong et al. [12] divided the welding task information into

- Management Information
- Welding Feature Information
- Component Information

A custom schema is defined in EXPRESS language to include all the information regarding the welding process like geometry, groove, weld quality, tolerances [12]. Using this method, all the information about the weld joint can be included in the STEP file. However, all the information is not required for the automation and control of robotic welding operation. For example, management information. The management information is not directly used in the control of robotic welding. This information can be made available at the system level like access control or included at the overall product level like product designer or approver. Rather than using/re-using the entities which are already defined under the STEP AP242 standard, Weidong et al. [12] defined a completely new schema. As an alternative, the following two methods will explain how welding information can be added to the STEP files using the entities already defined in the standard.

B. *Using Unicode Text*

Traditionally, welding information is included in the manufacturing drawing as a weld symbol. The weld symbol carries all the information needed to carry out the welding task in a graphical form. This weld symbol is attached to the weld seam of the 3D CAD model as part of MBD. Most commercial CAD systems treat this weld symbol as a presentation entity. So this weld symbol is not machine-readable and useful for the automation of welding tasks. This weld symbol is made semantic by using Unicode string based on the CAx recommendations of ”PMI Unicode String Specification Examples and Mapping Strategies” [20]. All the 39 symbols defined in AWS and ISO standards for welding annotations are represented using Unicode characters. Of these 39 symbols, 22 symbols are mapped to Unicode characters from various Unicode character sets. The remaining 17 symbols use a custom-defined code. Table I shows all the weld symbols and their Unicodes and custom codes used.

1) *Custom code for weld symbols*

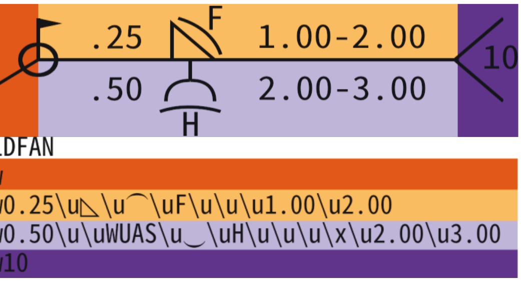
Not all weld symbols can be mapped to Unicode characters, as mentioned above. Custom codes are defined for the 17 symbols, which cannot be mapped to Unicode characters. The custom code ‘WXXX’ is a 4-letter combination starting with ‘W’ for ‘Weld’. The last two letters indicate if the weld symbol is on ‘Other Side’ or ‘Arrow Side’ using ‘OS’ or ‘AS’, respectively. The second letter indicates the type of symbol. When two letters (second and third) are needed for identifying the weld symbol, as in Flared V groove or Flared Bevel groove, then ‘O’ and ‘A’ are used to indicate ‘Other Side’ and ‘Arrow Side’, respectively. If there is no ‘Other Side’ or ‘Arrow Side’ to show, as in the case of ’Melt Through’, three letters are used for the symbol.

TABLE I. Welding Symbols and Their Unicode and Custom Codes

Welding Symbol	Description	Unicode	Custom Code
⏸	Sqare groove	23F8	-
Scarf	Scarf groove	-	WSCG
v	V groove - Other Side	2304	-
^	V groove - Arrow Side	2303	-
Bevel	Bevel groove - Other Side	-	WBOS
	Bevel groove - Arrow Side	-	WBAS
U	U groove - Other Side	-	WUOS
	U groove - Arrow Side	-	WUAS
J	J groove - Other Side	-	WJOS
	J groove - Arrow Side	-	WJAS
Flare V	Flare V groove - Other Side	-	WFVO

	Flare V groove - Arrow Side	-	WFVA
Flare Bevel	Flare Bevel groove - Other Side	-	WFBO
	Flare Bevel groove - Arrow Side	-	WFBA
◺	Fillet weld - Other Side	25FA	-
◸	Fillet weld - Arrow Side	25F8	-
Plug	Plug - Other Side	-	WPOS
	Plug - Arrow Side	-	WPAS
⎴	Slot - Other Side	23B4	-
⎵	Slot - Arrow Side	23B5	-
⊗	Stud	2A02	-
○	Spot or Projection - One Side	25EF	-
⦵	Spot or Projection - Both Sides	29B5	-
Seam	Seam - Other Sides	-	WSOS
	Seam - Arrow Sides	-	WSAS
	Seam - Both Sides	-	WSBS
◠	Back - Other Sides	25E0	-
◡	Back - Arrow Sides	25E1	-
	Surfacing	23D6	-
⊓	Edge - Other Sides	2293	-
⊔	Edge - Arrow Sides	2294	-
	All Round	232E	-
Melt Through	Melt Through	-	WMTH
◻	Consumable Insert - Square	25FB	-
▭	Backing - Rectangle	25AD	-
	Spacer - Rectangle	23DB	-
⎯	Flat Contour	23AF	-
⏜	Convex Contour	23DC	-
⏝	Concave Contour	23DD	-

Following [20], the welding symbol is encoded as a Unide string starting with WLD followed by three characters served for ‘Flag’, ‘All Round’, and ‘Reverse’. If these symbols e present in the weld annotation, the corresponding first letters , A and R) are added. Otherwise, ‘N’ is added in the rresponding location.



```
DFAN

0.25\u◺\u⏜\uF\u\u\u1.00\u2.00
0.50\u\uWUAS\u◡\uH\u\u\u\x\u2.00\u3.00
10
```

. 2. Regions of a welding symbol and the corresponding regions hlighted in the Unicode string. The unicode string is split into multiple lines clarity [20]

A welding symbol is separated into 4 different regions as can seen in Fig. 2, and the regions are separated using ‘\w’ in the ing. Within a region, values are separated by ‘\u’, and ‘\x’ is used to indicate number of seams and similar information. As per Part 21 of the STEP standard [21], the Unicode characters in a string shall be placed between ‘\X2\’ and ‘\X0\’ to denote their start and end, respectively. Fig. 3 shows a T-joint weld with its welding annotation.

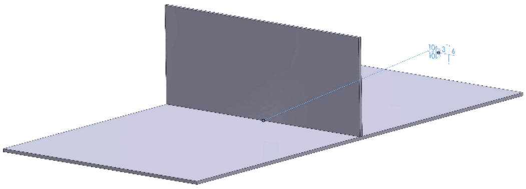


Fig. 3. T-Joint with welding annotation

This symbol represents a fillet weld of size 10.0 on both sides. But the other side has intermittent welds of length 3.0 and pitch 6.0. The corresponding Unicode string for this weld is ”WLDNNN\w\w10.0\u\u\X2\25FA\X0\\u\X2\23DC\X0\\u\u\u\u3.0\u6.0\w10.0\u\u\X2\25F8\X0\\u\X2\23DC\X0\\u\u\u\x\u\u\w”. This Unicode string is included in the STEP file using ‘TEXT_LITERAL’ entity as shown below.

#14=TEXT_LITERAL(’Welding Annotation’, ’WLDNNN\w\w10.0\u\u\X2\25FA\X0\\u\X2\23DC\X0\\u\u\u\u3.0\u6.0\w10.0\u\u\X2\25F8\X0\\u\X2\23DC\X0\\u\u\u\x\u\u\w’, $, $, $, $);

This welding annotation is attached to the weld seam using ‘SHAPE_REPRESENTATION’ entity. This entity relates the welding annotation to the geometry of the weld seam i.e. the edge of the part. The particular instance of ‘SHAPE_REPRESENTATION’ can be identified by the description ‘Welding Annotation’. Once this instance is identified, the welding annotation and the weld seam can be readily extracted. The Unicode string is parsed to identify the welding parameters and passed to the task definition for automation and control of robotic welding. This method is easy to implement and uses the already defined entities from the STEP AP242 standard. This also aligns with the traditional practices of manufacturing drawing using AWS and ISO standard weld symbol and carries all the information defined in the weld symbol. This method can be implemented using the first edition of STEP AP242 standard.

C. *Using the Entities from STEP AP242 Ed2*

As stated above, the second edition of STEP AP242 has the capabilities to include the welding information in the STEP file. This method proposes to use these new structures to add and reuse the welding information for robotic welding. ‘Mating Capability’ domain of the standard will be used along with other entities to include the welding information. The following entities are used for this purpose.

- ASSEMBLY_SHAPE_CONSTRAINT_ITEM_RELATIONSHIP
- ASSEMBLY_SHAPE_CONSTRAINT
- ASSEMBLY_SHAPE_JOINT_ITEM_RELATIONSHIP

- ASSEMBLY_SHAPE_JOINT
- MATING_DEFINITION
- MATED_PART_RELATIONSHIP

The above entities in combination with other entities like 'SHAPE_REPRESENTATION' are used to add the welding information. The welding information which cannot be included directly using the entities from the second edition of the standard can be added by:

1. Using Unicode strings as described in the second method,
2. Defining custom parameters or entities for the missing information.

D. *Extracting the Weld Seam*

The most important step in robot programming for automated welding is locating the weld seam. The weld seam can be identified easily from STEP AP242 files. The welding annotations are attached to the seam. The geometry of the seam, its location with respect to the assembly coordinate system can be extracted from the STEP AP242 files as per the method described by Shafi et al. [4]. Fig. 4 shows the steps involved in identifying the seam location using the methodology described by Shafi et al. [4].

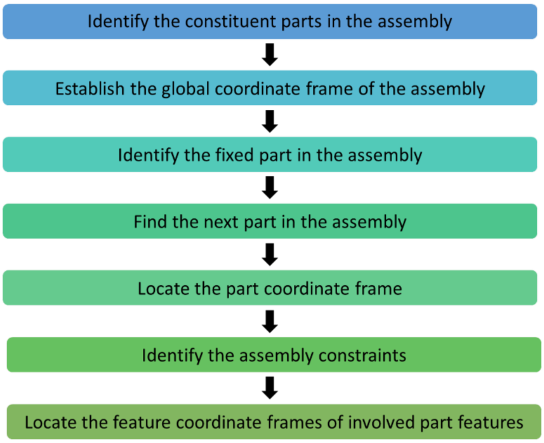

Identify the constituent parts in the assembly
Establish the global coordinate frame of the assembly
Identify the fixed part in the assembly
Find the next part in the assembly
Locate the part coordinate frame
Identify the assembly constraints
Locate the feature coordinate frames of involved part features

Fig. 4. Steps involved in identifying the seam [4]

E. *Using Tolerancing Information*

The tolerance information is available in the STEP files of constituent parts of the assembly, as mentioned earlier. This information can be used to perform the tolerance stack-up analysis of the assembly. This gives the tolerances applicable to the weld seam, i.e. the maximum allowable weld seam deviation from the defined value. This can be used as an input for control of robotic welding operation in combination with real-time sensor information.

VI. Task Specification Using The Constraint Information

The welding task is defined by converting the information extracted from STEP AP242 files into position, orientation, and motion constraints on the manipulator end-effector/Tool Center Point (TCP). A brief description of various constraints on TCP is given below.

- Position: The position of TCP is determined by the geometry of the parts, their mating constraints, and the welding process. The welding process limits the minimum and maximum vertical distance of TCP from the weld surface (axis). The position of the weld can be found from critical dimensions and GD&T.
- Orientation: The orientation of the tool is expressed in the travel angle and work angle. These angles depend on the weld type and the geometry of the parts. This gives the tool three rotational degrees of freedom (DoF), one full DoF about the tool axis, and two DoFs limited by travel and work angles.
- Motion: The direction of motion is determined by the geometry and positioning of the parts. The velocity of motions is mainly dependent upon the welding process. The tool has two degrees of freedom in translation, one DoF along the weld axis and another limited translation along the vertical to weld axis.
- Weld Length: The length of weld and pitch (center-to-center distance) of welds can be extracted from welding symbols, which define the extent of the tool motion.

VII. Implementation

The presented work is implemented [22] in Java using the JSDAI API [23] in the Eclipse development environment [24]. JSDAI is a Java based API for the Standard Data Access Interface (SDAI) [25]. It is used to read and write STEP files or any EXPRESS based data model.

The JSDAI Express Compiler [26] is used to parse and generate Java classes for entities defined in STEP AP242 Ed2 [16]. Java programs have been developed to add and extract welding annotations in STEP files. Moreover, a parser has been implemented to extract the welding parameters from Unicode welding annotation string. This parser uses the weld symbols defined in an XML file.

VIII. Conclusion

This paper has discussed the possibility of extracting constraint information from STEP AP242 files and use it for task specification for welding operations. All the relevant information for welding processes is identified, and how this can be extracted from STEP AP242 files is explained. This will enhance the control of the manipulator during the welding process and improve the weld quality. The proposed method of including welding information using a Unicode string, or AP242 Edition 2 mating entities is compliant with existing Express parsing tools such as JSDAI. An example implementation is given for JSDAI, and the information was successfully extracted. Future work includes integration with eTaSL and similar constraint-based robot programming software.

REFERENCES

[1] J. Norberto Pires, Altino Loureiro, Gunnar Bolmsjo, Welding Robots: Technology, System Issues and Applications. Springer, 2006.

[2] Tom Whitter, “Best Practices for Robotic Welding,” Manufacturing Engineering, vol. 153, pp. 77–84, 2014.

[3] Ed Sinkora, “New Technologies Drive Growth of Robotic Welding,” Advanced Manufacturing, pp. 73–80, October, 2017.

[4] Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad, “Leveraging Model Based Definition and STEP AP242 in Task Specification for Robotic Assembly,” Procedia CIRP, vol. 97, pp. 92-97, 2021. [Online]. Available: https://doi.org/10.1016/j.procir.2020.05.209

[5] L. I. Lieberman, M. A. Wesley, “AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly,” IBM Journal of Research and Development, vol. 21, no. 4, pp. 321–333, 1977.

[6] Stephen G. Kaufman, Randall H. Wilson, Rondall E. Jones, Terri L.Calton, “The Archimedes 2 Mechanical Assembly Planning System,” Proceedings of IEEE International Conference on Robotics and Automation, vol. 4, pp. 3361–3368, 1996.

[7] U. Thomas, F. M. Wahl, “A System for Automatic Planning, Evaluation and Execution of Assembly Sequences for Industrial Robots,” Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, pp. 1458–1464, 2001.

[8] Joris De Schutter, Tinne De Laet, Johan Rutgeerts, Wilm Decre, Ruben Smits, Erwin Aertbelien, Kasper Claes, Herman Bruyninckx, “Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty,” International Journal of Robotic Research, vol. 26, no. 5, pp. 433–455, 2007.

[9] Erwin Aertbelien, Joris De Schutter, “eTaSL/eTC: A constraint-based Task Specification Language and Robot Controller using Expression Graphs,” IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1540–1546, 2014.

[10] J. Norberto Pires, T. Godinho, P. Ferreira, “CAD Interface for Automatic Robot Welding Programming,” Industrial Robot: An International Journal, vol. 31, no. 1, pp. 71–76, 2004.

[11] Nathan Larkin, Andrew Short, Zengxi Pan, Stephen van Duin, “Automated Programming for Robotic Welding,” Transactions on Intelligent Welding Manufacturing, pp. 48–59, 2020.

[12] Weidong Shen, Tianliang Hu, Chengrui Zhang, Yingxin Ye, Zhengyu Li, “A welding task data model for intelligent process planning of robotic welding,” Robotics and Computer Integrated Manufacturing,vol. 64, 2018.

[13] J. B. Herron, Re-Use Your CAD: The Model-Based CAD Handbook. Create Space Independent Publishing Platform, 2013.

[14] ISO 10303-203:2011, “Industrial automation systems and integration—Product data representation and exchange - Part 242: Application protocol: Configuration controlled 3D design of mechanical parts and assemblies,” International Organization for Standardization, Geneva, Switzerland, 2011.

[15] ISO 10303-214:2010, “Industrial automation systems and integration—Product data representation and exchange - Part 214: Application protocol: Core data for automotive mechanical design processes,” International Organization for Standardization, Geneva, Switzerland, 2010.

[16] ISO 10303-242:2020, “Industrial automation systems and integration—Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering,” International Organization for Standardization, Geneva, Switzerland, 2020.

[17] ASME Y14.41:2019, “Digital Product Definition Data Practices,” The American Society of Mechanical Engineers, 2019.

[18] AWS A2.4:2012, “STANDARD SYMBOLS FOR WELDING, BRAZING, AND NONDESTRUCTIVE EXAMINATION,” American Welding Society, 2012.

[19] ISO 2553:2013, “Welding and allied processes — Symbolic representation on drawings — Welded joints,” International Organization for Standardization, Geneva, Switzerland, 2010.

[20] CAx Implementor Forum, “PMI Unicode String Specification Examples and Mapping Strategies - for Dimensioning and Tolerancing, GD&T, Surface Texture Symbol, and Welding Symbol PMI Annotation Entities Revision,” vol. Revision J, 2011.

[21] ISO 10303-21:2016, “Industrial automation systems and integration—Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure,” International Organization for Standardization, Geneva, Switzerland, 2016.

[22] Shafi K. Mohammed, Welding-Annotations-for-Robotic-Welding, Available: https://github.com/mohshaf/Welding-Annotations-for-Robotic-Welding.git

[23] LKSoftWare GmbH, Germany. Jsdai. [Online]. Available: https://www.jsdai.net, Accessed: April 30, 2021

[24] Eclipse Foundation. Eclipse. [Online]. Available: https://www.eclipse.org/, Accessed: April 30, 2021

[25] ISO 10303-22:1998, “Industrial automation systems and integration — Product data representation and exchange - Part 22: Implementation methods: Standard data access interface,” International Organization for Standardization, Geneva, Switzerland, 1998.

[26] LKSoftWare GmbH, Germany. Jsdai express compiler. [Online]. Available: https://www.jsdai.net/core/express-compiler, Accessed: April 30, 2021

A.3. Paper III:

Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad. "Using Semantic Geometric Dimensioning and Tolerancing (GD&T) Information from STEP AP242 Neutral Exchange Files for Robotic Applications," (International Journal on Interactive Design and Manufacturing (IJIDeM)).

International Journal on Interactive Design and Manufacturing (IJIDeM)
https://doi.org/10.1007/s12008-023-01242-7

ORIGINAL PAPER

Using semantic Geometric Dimensioning and Tolerancing (GD&T) information from STEP AP242 neutral exchange files for robotic applications

Shafi Khurieshi Mohammed[1] · Mathias Hauan Arbo[2] · Lars Tingelstad[1]

Received: 7 September 2022 / Accepted: 13 February 2023

Abstract
This article proposes the use of Geometric Dimensioning and Tolerancing (GD&T) information from STEP AP242 for robotic manufacturing applications. This information can be directly added to the relevant features of the 3D model as per Model Based Definition methodology during the design phase of the product life cycle. STEP AP242 neutral exchange files enable the availability of product definition at the downstream operations, thus completing the Digital Thread as part of Industry 4.0 practices. This article discusses two methods; using custom Unicode strings and the standard entities; for including GD&T using the STEP AP242 Edition 2 neutral file exchange format. A method to form a single Unicode string to add all the GD&T information to the STEP files is described in this paper. The GD&T information in the Unicode string is fully semantic and can easily be parsed to extract the relevant PMI for tolerance analysis. A novel process of extracting and interpreting the relevant PMI for robotic manufacturing applications is described in detail. This article discusses various applications of this information for robotic manufacturing through two use-cases using the second edition of the STEP AP242 standard.

Keywords Robot · Assembly · Welding · STEP AP242 · ISO 10303 · Model based definition · Product manufacturing information · PMI · Geometric dimensioning and tolerancing · GD&T

Abbreviations

AM	Application Module
AP	Application Protocol
API	Application Programming Interface
ASME	American Society of Mechanical Engineers
ATI	Advanced Technologies for Industry
AWS	American Welding Society
CAD	Computer Aided Design
DT	Digital Thread
FCF	Feature Control Frame
GD&T	Geometric Dimensioning and Tolerancing
ISO	International Standards Organisation
JSDAI	Java Standard Data Access Interface
MBD	Model Based Definition
MBE	Model Based Enterprise
NC	Numerical Control
OEM	Original Equipment Manufacturer
PMI	Product and Manufacturing Information
QIF	Quality Information Framework
SME	Small and Medium Enterprise
STEP	STandard for Exchange of Product Data

1 Introduction

Robotics is one of the 16 Advanced Technologies for Industry (ATI) identified by the European Commission. The successful implementation of Industry 4.0 depends on the close integration of design and manufacturing by establishing a Digital Thread (DT) of product and process data which facilitates the automation of industrial operations. The use of industrial robots is increasing in automotive and electronics industries [1,2] much faster than in other industries. The main reason for this trend is that the product variation is low in these industries, and the production volumes are high. This reduces the cost of automation and increases productivity, thus providing more significant rewards for using robotics.

Shafi Khurieshi Mohammed
shafi.k.mohammed@ntnu.no

[1] Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology (NTNU), Trondheim 7034, Norway

[2] SINTEF Manufacturing AS, Trondheim 7031, Norway

Published online: 21 April 2023

The adoption of robotic automation in the case of Small and Medium Enterprises (SMEs) is less than half compared to that of large enterprises [3]. Generally, SMEs handle a variety of products in small volumes, which necessitates changing the robot program every time the product changes. This increases the cost of automation and reduces the production time, hence less use of robots in SMEs. The time and cost of robot programming can be reduced by using the product information from the design phase for automated programming. The establishment of a DT enables product data sharing to the downstream manufacturing operations. Some robot programming research use CAD software to define the motions relative to the product design such as Autopass [4], HighLap [5,6], Archimedes 2 [7], or more recently the work of Perzylo et al. [8], and Pane et al. [9]. The CAD data is used to associate end-effector motions or control strategies based on the relative geometric dimensions in the design. Mohammed et al. [10] used the assembly constraints from STEP AP242 files for robotic assembly operations to exemplify a direct connection between robot programming and the neutral exchange format.

Another aspect that limits the use of industrial robots is uncertainty. The uncertainties can be grouped as product, manipulator, and environment or work-cell uncertainties. The effects of these uncertainties can be mitigated by calibration of robots, using sensors, active or passive compliance, and careful placement of parts and robots in the work-cell. These approaches require process-specific tuning and calibration, often stemming from the experiential knowledge of the process planner or automation provider, or are arrived at by rigorous experimentation with real parts. This represents a disconnect between the design process and the downstream manufacturing process.

Traditionally the product uncertainties are handled by specifying the tolerance limits. The designers design the products to ideal size and shape to perform their function. However, it is impossible to manufacture the parts to exact nominal dimensions due to the limitations in manufacturing processes, as noted by Srinivasan [11]. Hence the designers specify some deviation from the nominal dimensions within which the product can perform its intended functions. These deviations from the nominal dimensions are known as part tolerances. As the part tolerances become tighter, the manufacturing cost increases, and if the tolerances become loose, the part may not function as expected.

While the tolerances on dimensions allow for variation in size, the Geometric Dimensioning and Tolerancing (GD&T) captures part variations in the size, form, and shape. GD&T is an important aspect of the Product and Manufacturing Information (PMI) which captures non-geometric product data. The variations in the individual parts affect the overall product assembly and the assembly process itself. These variations should be considered in robotic assembly for the planning, programming, and successful completion of the assembly tasks.

Proctor et al. [12] suggested the use of GD&T information for robotic manufacturing. They suggested that the accuracy of manipulators can be increased by using part tolerances for robotic path planning. It was also suggested that the GD&T information could be used for robot selection and part placement in work cells. Proctor et al. [13] used semantic GD&T and Quality Information Framework (QIF) to handle the uncertainty in robotic manufacturing and assembly operations.

This paper follows up on the idea by Proctor et al. [13] and discusses the use of GD&T information directly from the neutral CAD files as input to define robotic tasks. The STEP AP242 Ed2 file format is used for this purpose. The contributions of this paper are:

1. A proposal for the use of GD&T from design phase for robotic automation of assembly and welding operations,
2. A description of two methods to semantically add all the GD&T information to the STEP AP242 files.
3. A description of the process of extracting the relevant PMI for robotic applications.
4. A formation of a single Unicode string to capture the dimension, datums, Feature Control Frames, and other related PMI.
5. A presentation of use-cases where this GD&T information can be used for robotic applications.

This paper is structured as follows. Section 2 presents the basics of robot programming and the relevance of GD&T for robotic assembly and welding. Model-Based Definition (MBD) is presented in Sect. 3. Section 4 describes the STEP AP242 standard briefly. Section 5 explains the concepts of geometric dimensioning and tolerancing and its effects on assembly. Section 6 describes the methods of including the GD&T information in STEP AP242 files using custom Unicode string and using the entities defined in the STEP AP242 standard. Section 7 presents various use-cases for the applications of tolerancing information from STEP AP242 for robotic manufacturing and assembly processes. The concluding discussion is in Sect. 8.

2 Robot programming

The product data from annotated 3D CAD models can be used for robot programming and control for various manufacturing tasks like assembly and welding. The three traditional methods of robot programming are (1) teach-pendant programming, (2) offline programming, and (3) programming by demonstration [14]. As robots have high repeatability, both teach-pendant programming, where the technician jogs the robot to the desired positions, and programming by demonstration, where the technician physically guides the robot through the process, utilize the repeatability of modern industrial robots [15]. Most industrial robots are capable of repeating a motion with high accuracy. This is because the errors caused by varying end-effector payload, inverse kinematics deviations, or link and gear flex are sufficiently deterministic [16–18]. For offline programming, the precise calibration of both the robot and its environment is required to keep a direct correspondence between the programmed scenario and the physical setup [19]. In both cases, the repeatability and calibration errors can be handled using sensor-feedback control strategies.

Procter et al. [13] describe a method to combine and evaluate the uncertainty of the robot kinematics and the GD&T information to make better choices regarding which robot hardware to use and to improve or verify motion plans. Pane et al. [9] describe extracting sensor-feedback control strategies from CAD constraints and suggest inferring control gains based on GD&T information. These are examples of how a direct connection between the design process and programming of the automation process can be of value. The GD&T information can also potentially serve a purpose during the deployment of an automation process by evaluating the source of failures during production. For production lines with various part designs, providing information about which features the robot is expected to have interacted with during a failure, and their GD&T information can allow for faster failure-recovery and evaluation of the source of the error.

In many SMEs, a different company often provides robotic automation solutions that may use different CAD vendors and software solutions than the SME itself. This means that the GD&T information should be provided in a neutral exchange format to facilitate its usage in robotic automation solutions.

2.1 Relevance of GD&T in robotic assembly

The two essential parameters of any insertion operation are clearance and insertion length. Clearance is also represented as the clearance ratio, and it is defined as the ratio of the clearance to hole diameter [20]:

$$r_c = (d_h - d_p)/d_h \tag{1}$$

where r_c is the clearance ratio, d_h is the hole diameter, and d_p is the peg diameter.

Part clearances, clearance ratio, and insertion lengths can be used to estimate the contact states and force needed to complete the assembly. The clearance ratio also determines the maximum permissible tilt during the assembly process. Simunovic [21] used the part dimensions and insertion lengths to estimate the type of contact and insertion forces during assembly. Whitney [20] used the clearance ratio and insertion lengths to estimate the contact states, forces and jamming during peg-in-hole assembly. Clearance also determines the maximum tilt and the conditions of jamming in a chamferless assembly as described by Haskiya et al. [22]. The applicability of contact and force models also depends on the part clearances [23].

Part clearances also determine the suitability of the robotic manipulator and the success of the assembly operation. It also determines the control methodology needed for completion of the assembly [24,25]. ElMarghy et al. [26] discussed the effects of part dimensions and tolerances along with manipulator accuracy and repeatability on the success of assembly operation. As the uncertainty-to-clearance ratio increases, the completion of assembly becomes difficult with position control alone [27,28]. Hence depending on the clearance, an appropriate control strategy can be selected [29]. As the clearance ratio decreases, the success of assembly decreased [28,30].

The extraction of part dimensions and tolerances from the STEP files is used to calculate the effective clearance, which can be used to estimate the contact states and possibility of jamming; the assembly forces needed; decide the suitability of a manipulator for the assembly task; and to decide the control strategy.

Once the toleranced features and the tolerance values are extracted, these values are used to calculate the variation of these features from the nominal values. These values are used for defining the constraints for the robotic assembly tasks. Automated tolerance analysis programs can calculate the variation of toleranced features and their propagation in the entire assembly. Effective (worst possible) clearances between mating parts can be evaluated using these tolerance values. Considering the size tolerances, the effective clearance can be calculated as [26]

$$c_e = ((d_h + t_h) - (d_p + t_p))/2 \tag{2}$$

where c_e is the effective clearance, t_h is the tolerance on hole diameter, and t_p is the tolerance on peg diameter. Using Monte-Carlo simulations and other automated computer programs, the tolerance information can be analyzed to describe the clearances that can be expected.

Many designs have multiple features that can come into contact during the assembly operation. Providing the complete design with PMI makes it possible to evaluate the assembly process to determine which features are relevant to the process. For robot programs, the change in relevant features can require a change in control strategy required depending on the tolerance values.

2.2 Relevance of GD&T in robotic welding

PMI in general will be very useful in the automation of welding process. The PMI along with product geometry can be used for automatic programming of welding robots. This will decrease the dependence on costly and time consuming manual programming and increases its adoption in SMEs where changing product design requires frequent changes in robot programs. Mohammed et al. [31] discussed the importance of PMI in robotic welding and described the reuse of welding information for STEP AP242 files. In addition to welding annotations, GD&T information will also be useful in automated robotic welding.

The quality and the success of the weld depends on the root gap and the fit–up of the components being welded. These two parameters depend on the seam geometry, edge preparation and material thickness. The geometric variations of the parts being welded also effect the seam position, orientation and geometry, thus affecting the weld path, positioning and orientations of weld gun. The effects of these variations are limited by specifying various tolerances. The tolerances that impact the weld quality can be divided into two groups.

- Groove tolerances: The tolerances specified for edge preparation are considered as groove tolerances. These are applied to root gap, bevel angles and root face. *AWS D1.1 Structural Welding Code – Steel* gives the tolerances for welding of steel structures. *ASME B16.25–2017: Buttwelding ends* standard specifies the edge preparation and tolerances for buttwelding of piping components. These tolerances can be easily included in the Unicode welding annotations described by Mohammed et al. [31].
- General tolerances: The tolerances specified on the welding components that affect the seam geometry, position and fit–up are considered in this group. The standard *ISO 13920:1996 – Welding – General tolerances for welded constructions – Dimensions for lengths and angles – Shape and position* specifies the size and positional tolerances for welded structures.

In manual welding, the welder senses the variations and errors and makes adjustments to successfully weld the parts. The product variations should be taken into account in robotic welding processes to avoid product quality issues and rejections. Generally in robotic welding, sensors are used to scan and extract the seam and groove geometry [32]. The sensing operation is planned based on the human input. This can be improved using the information from the CAD files. This paper demonstrates how the groove tolerances can be included in the welding annotations.

3 Model based definition (MBD)

The designers define a product and create 3D product models using commercial CAD software. Then the 2D manufacturing drawings are prepared from the 3D CAD models by adding the necessary PMI. Therefore, the 2D manufacturing drawing becomes the master and source of product information, and these drawings are the input to the downstream manufacturing operations. Consumption of information from 2D drawings at downstream operations requires manual intervention to identify and recreate the necessary data for that specific operation. Recreating data at each downstream operation is time-consuming and may introduce human errors. This involvement may lead to product quality issues and delays as a result of miscommunication between the design and the manufacturing teams. These problems can be avoided by reusing the product data from the design phase in the downstream manufacturing operations. The manual recreation of information from 2D drawings is eliminated by adopting a Model Based Definition (MBD) methodology.

In MBD, the PMI is semantically added to the relevant features of the 3D CAD model during the design phase. Then the 3D model carries the critical dimensions, GD&T, surface finish, and other needed information along with the product geometry, making it the master of product information [33,34]. Figure 1 shows a 3D CAD model with all the PMI attached to its features. These annotated 3D models are the basis of connected DT in a Model Based Enterprise (MBE). The product information will be available to the downstream operations for direct use like NC machining and automated inspection [35]. Many contemporary applications of MBD are mentioned by Goher et al. [36].

The MBD created by the designers is shared with the downstream operators and subcontractors. Designers use commercial CAD software to prepare the 3D models and MBDs. Sharing the MBDs in native CAD formats is not feasible and is not a good practice as this requires the same commercial CAD software. Not all the stakeholders have access to the CAD software, nor are they knowledgeable about using that software. Most subcontractors are SMEs who serve various original equipment manufacturers (OEMs). They cannot purchase the licenses of all the CAD software used by different OEMs. Hence a neutral exchange format is needed, facilitating the sharing of product designs to all stakeholders avoiding the dependency on commercial CAD software in the downstream operations. STEP AP242 is such a neutral exchange file format that can carry both the product geometry and PMI. The following section describes this exchange file format.

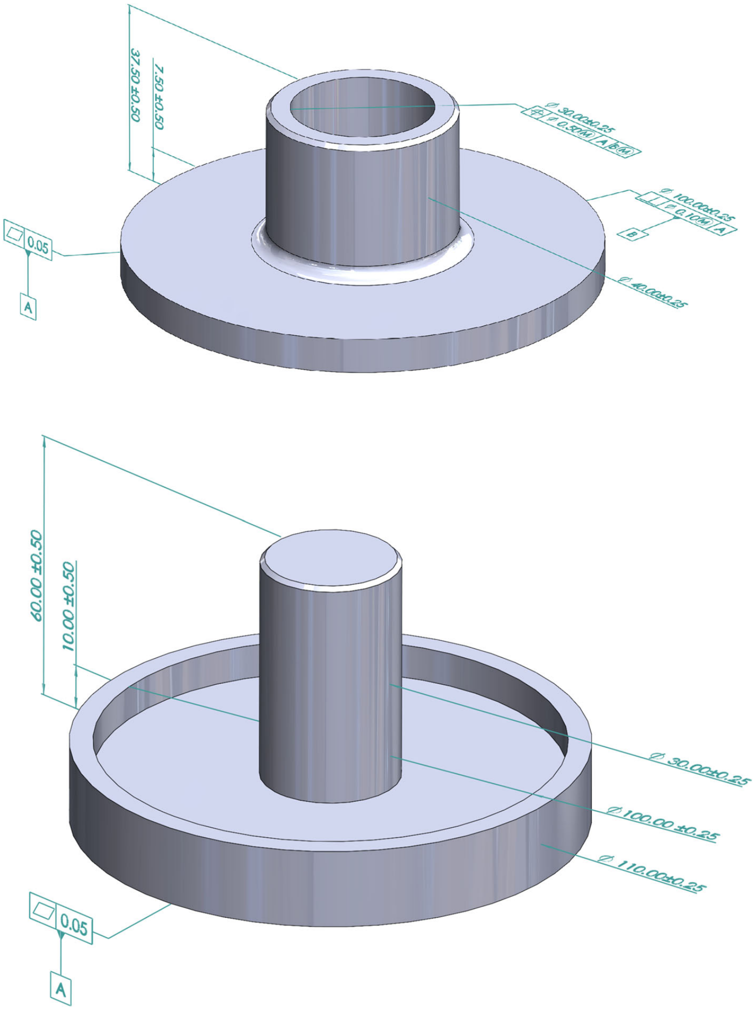


Fig. 1 Parts with GD&T annotations

4 STEP AP242

STandard for the Exchange of Product Model Data (STEP) is one of the industry's most widely used neutral file formats. STEP AP242 is part of a family of ISO (International Standards Organization) standards *10303: Automation systems and integration – Product data representation and exchange*. The *AP242: Managed model–based 3D engineering* is one of the latest application protocols that deals with 3D semantic PMI [37]. The latest (second) edition of this Application Protocol (AP) was released in 2020 [38]. STEP AP242 replaced AP203 (Configuration controlled 3D designs of mechanical parts and assemblies) and AP214 (Core data for automotive mechanical design processes). The AP203 and AP214 were developed for aerospace and defense and automotive industries respectively [39,40]. Using STEP AP242, MBD can be created as per *ASME Y14.41–2019: Digital Product Definition Data Practices* [41]. The use of STEP AP242 in the industry increases with increasing digitization. Many researchers are working on the capabilities of STEP AP242 and its applications in industry like automated tolerance analysis [42], CAD/CAE integration, and smart manufacturing [43–45].

STEP AP242 allows the addition of PMI both semantically and non-semantically. The non-semantic addition of PMI is for presentation purposes and not relevant for robot programming, and only semantic elements are considered for PMI addition and extraction for constraint definition. The following information is available in STEP AP242 files.

Product Geometry The exact product geometry is translated into STEP, and it can be readily extracted and used to identify part features.

Product Structure and Assembly Information STEP AP242 files also carry the product structure in terms of the constituent parts of the assembly and their relative positions. Using the Mating Capability Domain Model of STEP AP242 Ed.2, mating and joint type can also be included in the assembly files.

Critical Dimensions and GD&T The dimensions critical for quality and the GD&T can be semantically added to the part features in a STEP AP242 file. These can be used to extract the part features' sizes and locations and estimate possible deviation from the nominal design values.

Annotations Other PMI like surface finish, thread specifications, and welding symbols can be included in the STEP files in the form of annotations. Mohammed et al. [31] presented two different ways in which the welding annotations can be added to the STEP AP242 files.

Properties Product properties like material, weight, and envelope dimensions can be extracted from the STEP AP242 files. These can be used to estimate the automation of handling and packaging.

Notes Other relevant information can also be included in the MBD in the form of text annotations. These may include the special tools, fixtures, and essential process-related information which can be used to define additional constraints on the manipulator.

4.1 Structure of STEP files

A STEP file is a text-based file that encapsulates all the product details in entities. These files are called Part-21 files because this text encoding is done as per Part-21 of ISO 10303 standard [46]. These files are also known as physical files and have the extension '.STP' or '.STEP'. These files are human-readable and facilitate the writing, reading, and exchange of product data. The significant entities used for representing product geometric information are

Geometric Entities The sub-classes of 'geometric_representation_item' are considered as geometric entities.

Topological Entities The sub-classes of 'topological_representation_item' are considered as topological entities. There are some entities like 'vertex_point' and 'edge_curve' which are sub-classes of both 'geometric_representation_item' and 'topological_representation_item'. Such type of entities are also considered as topological entities.

Presentation Entities The entities from 'Presentation appearance' schema, along with the entities from other schemas of 'Visual appearance (Part: 46)' are used to define the graphical appearance of the model. These entities are considered as 'presentation entities'.

Representation Entities Entities from the schemas of 'Fundamentals of product description and support (Part: 41)' are considered as representation entities. These entities are used to identify the product and classify and establish relationships among products. Various sub-classes of 'representation' and 'representation_relationship' entities, which are used to establish relationships between different entities, are also included in this category.

Figure 2 shows the different types of entities and their relationships for a rectangular block. The geometric entities represent the basic geometric elements of product design like a Cartesian point, a line, or a plane. These entities are combined or referred to in topological entities to form the features of the product geometry like a vertex, an edge or a face of the product. The higher entities bind all the topological entities and represent the entire product, for example, the 'Mechanical_Design_Geometric_Presentation_Representation' entity.

Where possible, the annotations are attached to the topological entities to refer to the part features that are affected by these annotations. This is also suggested by the CAx–IF recommendation [47], where it recommends using the 'advanced_face' entity for part features while attaching the GD&T annotations. This helps identify the part features quickly and enhances the reuse of product data in the downstream operations.

5 Geometric Dimensioning and Tolerancing (GD&T)

The practice of GD&T is mainly based on the set of standards from ISO and *ASME Y14.5: Dimensioning and Tolerancing standard* [48]. The standard ISO 1101 [49] deals specifically with form, orientation, location, and run-out tolerances. As the part features deviate in form, orientation, location, and run-out from the ideal values, they affect the assembly process also.

Each of these tolerances defines a tolerance zone that limits the deviation of the part feature from its ideal shape, size, and position. The shape of the tolerance zone depends on the characteristics of tolerance, and the size depends on the tolerance value. Table 1 gives the shape of tolerance zones defined by the type of the tolerance.

Figure 3 shows a part with flatness tolerance and the resultant tolerance zone. The flatness tolerance on the top surface of the part means that all the points on that surface lie within the tolerance zone formed by two parallel planes, separated by a distance equal to the tolerance value. The effects of the tolerances vary depending on how they are attached to the feature.

Fig. 2 Structure of STEP file: geometric, topological and representation entities

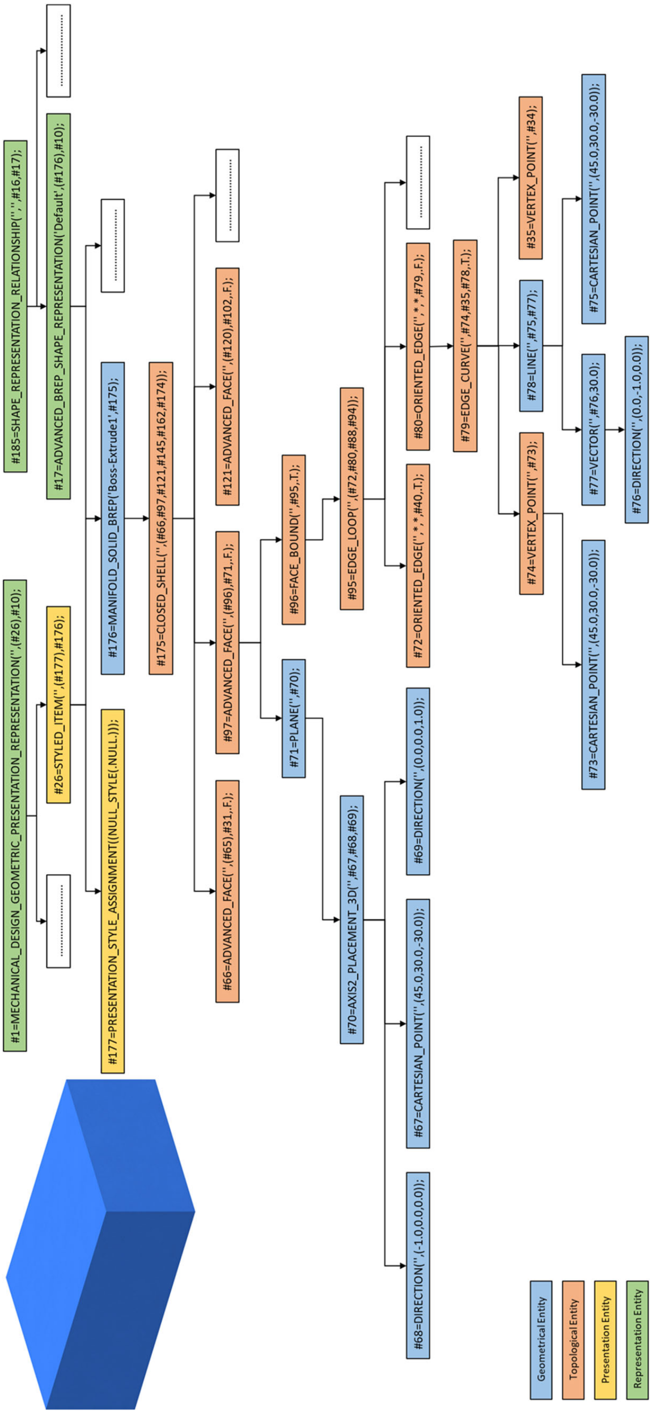

#185=SHAPE_REPRESENTATION_RELATIONSHIP('','',#16,#17);
#17=ADVANCED_BREP_SHAPE_REPRESENTATION('Default',(#176),#10);
#1=MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION('',(#26),#10);
#26=STYLED_ITEM('',(#177),#176);
#177=PRESENTATION_STYLE_ASSIGNMENT((NULL_STYLE(.NULL.)));
#176=MANIFOLD_SOLID_BREP('Boss-Extrude1',#175);
#175=CLOSED_SHELL('',(#66,#97,#121,#145,#162,#174));
#121=ADVANCED_FACE('',(#120),#102,.F.);
#97=ADVANCED_FACE('',(#96),#71,.F.);
#66=ADVANCED_FACE('',(#65),#31,.F.);
#96=FACE_BOUND('',#95,.T.);
#71=PLANE('',#70);
#95=EDGE_LOOP('',(#72,#80,#88,#94));
#70=AXIS2_PLACEMENT_3D('',#67,#68,#69);
#80=ORIENTED_EDGE('',*,*,#79,.F.);
#72=ORIENTED_EDGE('',*,*,#40,.T.);
#69=DIRECTION('',(0.0,0.0,1.0));
#67=CARTESIAN_POINT('',(45.0,30.0,-30.0));
#68=DIRECTION('',(-1.0,0.0,0.0));
#79=EDGE_CURVE('',#74,#35,#78,.T.);
#35=VERTEX_POINT('',#34);
#78=LINE('',#75,#77);
#74=VERTEX_POINT('',#73);
#73=CARTESIAN_POINT('',(45.0,30.0,-30.0));
#75=CARTESIAN_POINT('',(45.0,30.0,-30.0));
#77=VECTOR('',#76,30.0);
#76=DIRECTION('',(0.0,-1.0,0.0));
Geometrical Entity
Topological Entity
Presentation Entity
Representation Entity

Table 1 Geometric tolerances and their tolerance zones

Type	Characteristic	Symbol	Modifier/element applied to	Tolerance zone
FORM	Straightness	—	None	Two parallel lines
	Flatness	⏥	None	Two parallel planes
	Circularity	○	None	Two concentric circles
	Cylindricity	⌭	None	Two concentric cylinders
PROFILE	Profile of a Line	⌒	None	Two parallel lines in the cross-section
	Profile of a Surface	⌓	None	3D Volume along the feature
ORIENTATION	Angularity	∠	None	Two parallel planes
			Diameter	Cylindrical
	Perpendicularity	⊥	None	Two parallel planes
			Diameter	Cylindrical
	Parallelism	∥	None	Two parallel planes
			Diameter	Cylindrical
RUNOUT	Circular Runout	↗	Applied to a surface of revolution	Two concentric circles
			Applied to a surface perpendicular to the axis	Two circle of equal diameter separated along the axis by a distance equal to the tolerance value
	Total Runout	⌰	Applied to cylindrical surfaces	Two coaxial cylinders
			Applied to a perpendicular to the axis	Two parallel planes separated by distance equal to the tolerance value
LOCATION	Position	⌖	None	Two parallel planes
			Diameter	Cylindrical
	Concentricity\Coaxiality	◎	None	Cylindrical
	Symmetry	⌯	None	Two parallel planes

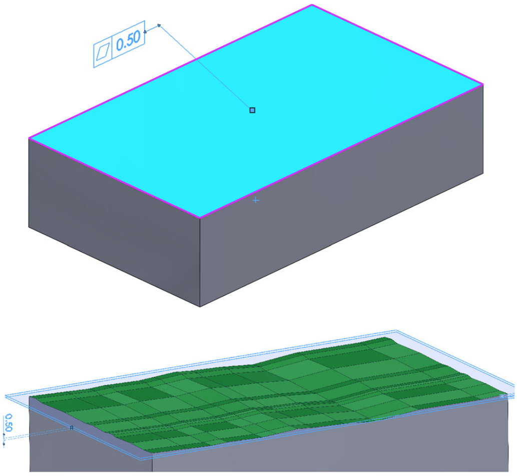


Fig. 3 Part with flatness tolerance and the resulting tolerance zone

1. If the Feature Control Frame (FCF) is attached directly to the feature, it applies to the feature surface.
2. If the FCF is attached to the dimension, it affects the feature's mid-plane, axis, or centerline.
3. For some characteristics like the profile of a line, if the FCF is attached to the surface or the edge of a surface in the perspective view, it is applied to the entire surface. But if it is attached to a line on a surface, it affects all the parallel line elements on that surface.

6 Including GD&T information in the STEP files

Most of the commercial CAD software support semantic annotation of GD&T in their native formats, even though there are some limitations [50]. Generally, the annotations do not retain their semantic property when saved as STEP AP242 files using most commercial CAD software. Except for two of the CAD software evaluated by Lipman and Filliben, all others need third-party converter tools to generate the STEP AP242 MBD files from the native CAD formats [51,52]. Below are some of the combinations in which the dimensions, FCFs, tolerances and datums can be represented on a 3D model:

1. Individual Dimension which can be basic, reference, or with tolerances. The tolerances can be unidirectional, bidirectional equilateral or bidirectional unequal.
2. Independent datum call outs
3. Independent FCFs
4. Dimensions with datum call outs
5. Dimensions with FCFs
6. Dimensions with both FCFs and datum call outs
7. FCFs with datum call outs

An example is shown in the annotations on the two components of a peg-hole assembly in Fig. 1.

This section describes two methods by which GD&T annotations can be semantically and directly added to the STEP AP242 files as per CAx recommendations [47,53]. JSDAI API [54], a Java based API for the Standard Data Access Interface [55], is used to edit the STEP AP242 files.

6.1 Using unicode text

The GD&T information and feature control frames can be included in the STEP files using Unicode strings based on the CAx recommendations of *PMI Unicode String Specification Examples and Mapping Strategies* [53]. All the GD&T symbols defined in ISO and ASME standards are represented using Unicode characters. Table 2 shows the Unicode and custom codes of GD&T symbols.

6.1.1 Forming the unicode string

A similar methodology is used as described by Mohammed et al. [31] in forming the Unicode string for GD&T annotation. Below are the key rules followed for this purpose.

1. The Unicode string starts with a six-letter combination 'GDTXXX'. Here the first three letters 'GDT' stand for 'Geometric Dimensioning and Tolerancing'.
2. The next three letters are used to indicate the dimensions, FCF, and datum callouts, respectively. The possible combinations of dimensions, FCFs and datum callouts are described in Sec. 6. Table 3 shows the letter codes used for this purpose. The letter 'N' is added in the corresponding location when any of these items are not present in the annotation.
3. Special symbols like all-round can be added after this six-letter combination using a region separator '\w' (see Point- 8 below).
4. The Unicode characters are placed between '\X2\' and '\X0\'.
5. For abbreviations, custom codes and identifiers, a 4–letter combination like 'TXXX' is used.
6. The first letter 'T' in the 4–letter combination denotes 'Tolerancing'.
7. To make all the values semantic, each and every value is separated as opposed to the CAx recommendation [53]. '\u' is used to separate different values.
8. '\w' is used to separate different types and regions.

The flag notes are not included in the Unicode string, as they can be attached to the part features as separate text annotations.

Figure 4 shows a feature control frame for a position tolerance attached to a dimension with equal bilateral tolerances. The following Unicode string captures the information in this annotation formed using the methodology described in this section.

```
’’GDTEFN\w\w\u\X2\2300\X0\\u30.00\uTPOM
\u0.25\w\u\X2\2316\X0\
\u\X2\2300\X0\\u0.50\u\X2\24C2\X0\\uA
\uB\u\X2\24C2\X0\’’.
```

This Unicode string is added to the STEP AP242 files using the entity ’TEXT_LITERAL’. ’SHAPE_ REPRESENTATION’ entity is used to associate this annotation with the corresponding part feature semantically.

6.2 Using the entities from STEP AP242 Ed2

All the GD&T information can be included in the STEP AP242 files using the entities defined in the standard. The methodology followed is given in CAx recommended practices for ”Representation and Presentation of Product Manufacturing Information (PMI) (AP242)” [47]. From this recommendation, only ’representation’ aspects are relevant as the information is semantically represented and reusable for robotic applications. This CAx recommendation uses the entities defined in the first edition of STEP AP242.

The authors of this article have analyzed the second edition of STEP AP242 and performed a comparative study against the first edition for the present work. The second edition has a set of changes in type and entity definitions from the first edition. However, these changes do not affect the methodology described in the CAx recommendation. Hence, the authors suggest using the same methodology described by CAx using the second edition of STEP AP242. The entities from the following Application Modules (AM) can capture the GD&T information of the parts: Dimension tolerance (Part: 1050), Geometric tolerance (Part: 1051), and Default tolerance (Part: 1052).

Table 2 Geometric dimensioning and tolerancing symbols, their unicode from CAx recommendation [53] and the custom codes

GD&T symbol	Description	Unicode	Custom code
∠	Angularity	2220	–
︵	Arc Length	FE35	–
↔	Between	2194	–
Ω	Capital Omega	03A9	–
℄	Center Line	2104	–
↗	Circular Runout	2197	
○	Circularity	25CB	–
◎	Concentricity	25CE	–
⌲	Conical Taper	2332	–
Continuous Feature	Continuous Feature	–	TCOF
Controlled Radius	Controlled Radius	–	TCTR
⌴	Counterbore	2334	–
⌵	Countersink	2335	–
⌭	Cylindricity	232D	–
⌀	Diameter	2300	–
Depth	Depth	–	TDPT
Envelope Requirement	Envelope Requirement	–	TEPR
⏥	Flatness	23E5	–
Ⓕ	Free State	24BB	–
Ⓘ	Independency	24BE	–
Ⓛ	Least Material Condition	24C1	–
Ⓜ	Max Material Condition	24C2	–
⫽	Parallelism	2AFD	–
⏊	Perpendicularity	23CA	–
Plus/Minus	Plus/Minus	–	TPOM
⌖	Position	2316	–
⌒	Profile of a Line	2312	–
⌓	Profile of a Surface	2313	–
Ⓟ	Projected Tolerance Zone	24C5	–
Radius	Radius	–	TRDS
Ⓡ	Reciprocity	24C7	–
Ⓢ	Regardless of Feature Size	24C8	–
⌳	Slope	2333	–
⍵	Small Omega	2375	–
Spherical Diameter	Spherical Diameter	–	TSPD
Spherical Radius	Spherical Radius	–	TSPR
Spot Face	Spot Face	–	TSTF
□	Square	25A1	–
Statistical Tolerance	Statistical Tolerance	–	TSTT
⏤	Straightness	23E4	–
⌯	Symmetry	232F	–
Ⓣ	Tangent Plane	24C9	–
⌰	Total Runout	2330	–
▷	Translation Modifier	25B7	–
Ⓤ	Unilateral/Unequally Disposed	24CA	–

Table 3 Keywords and their abbreviations [53]

Keyword	Abbreviation (identifier)
Basic Dimension	B
Reference Dimension	R
Dimension with Unidirectional Tolerances	S
Dimension with Equal Bidirectional Tolerances	E
Dimension with Unequal Bidirectional Tolerances	U
Feature Control Frame (ASME)/Tolerance Frame (ISO)	F
Datum Feature Symbol	D
Datum Target Symbol	T
AllAround Symbol	TAAS
All Over Symbol	TAOS
Derived Feature	TDFT

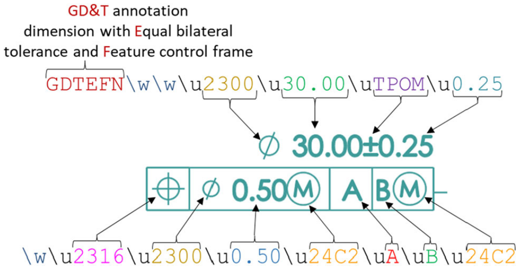

GD&T annotation
dimension with Equal bilateral
tolerance and Feature control frame
GDTEFN\w\w\u2300\u30.00\uTPOM\u0.25
∅ 30.00±0.25
⌖ ∅ 0.50Ⓜ A BⓂ
\w\u2316\u2300\u0.50\u24C2\uA\uB\u24C2

Fig. 4 GD&T annotation with the corresponding unicode string

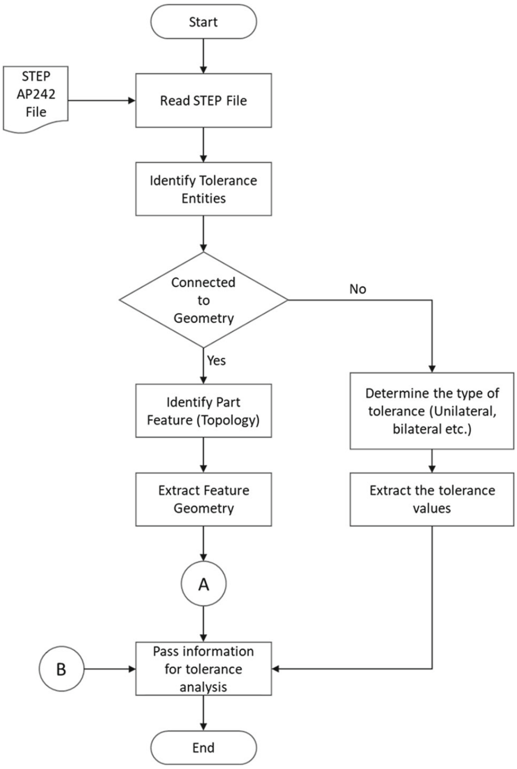

Start
STEP AP242 File
Read STEP File
Identify Tolerance Entities
Connected to Geometry
No
Yes
Identify Part Feature (Topology)
Determine the type of tolerance (Unilateral, bilateral etc.)
Extract Feature Geometry
Extract the tolerance values
A
B
Pass information for tolerance analysis
End

Fig. 5 Extraction of GD&T information from STEP AP242 entities

The entities from the first three AMs are mainly mapped to the entities of Integrated Generic Resource: Shape variation tolerances (Part: 47). The Part: 47 has three schemas that cover the information requirements for GD&T defined in these AMs. The schemas of Part: 47 are 'Shape aspect definition, Shape dimension and Shape tolerance'. Using the entities from these AMs and following the methodology of CAx recommendation, all the GD&T information can be added semantically to the part features.

Another useful AM is 'Extended geometric tolerance' (Part:1666). This AM specifies the entities for defining the tolerance zone boundaries and can be used for automated tolerance analysis and quality analysis.

6.3 Extracting the GD&T information

Extraction of GD&T information from the Unicode string is straightforward. From the 'SHAPE_REPRESENTATION' entity, the part feature and the corresponding Unicode tolerance annotation can be identified. The feature geometry is extracted using the standard geometric and topological entities of STEP AP242 standard, and the Unicode string is extracted from the entity 'TEXT_LITERAL'. A parser is developed to parse the Unicode annotation and extract the tolerance information. This information is used in forming the tolerance zones and tolerance analysis.

The overall process for extracting the tolerance information while using the standard entities from STEP AP242 requires more steps than the Unicode string. A flowchart of the process is shown in Fig. 5. After reading the STEP file, the program searches and identifies the tolerance entities. These are the entities from the schemas of Shape variation tolerances (Part: 47) of the standard. These entities are checked to determine whether these tolerances are connected to the geometry of part features. If the tolerance entities refer to 'SHAPE_ASPECT' or 'COMPOSITE_GROUP_SHAPE_ASPECT,' then these tolerances are attached to the part features. This corresponds to the path 'yes' after the 'Connected to Geometry' check in Fig. 5. In this case, the corresponding part feature and its geometry are extracted from the topological and geometric entities.

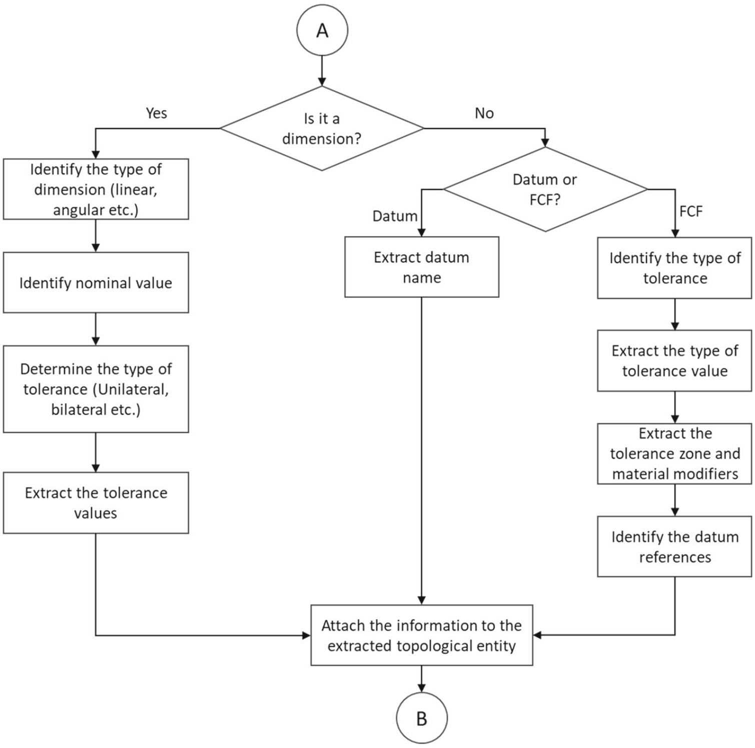

A
Is it a dimension?
Yes
No
Identify the type of dimension (linear, angular etc.)
Identify nominal value
Determine the type of tolerance (Unilateral, bilateral etc.)
Extract the tolerance values
Datum or FCF?
Datum
FCF
Extract datum name
Identify the type of tolerance
Extract the type of tolerance value
Extract the tolerance zone and material modifiers
Identify the datum references
Attach the information to the extracted topological entity
B

Fig. 6 Identification and extraction of dimension and geometric tolerances and datum features

The default tolerances are not attached to any particular geometric feature, hence do not refer to 'SHAPE_ASPECT' or 'COMPOSITE_GROUP_SHAPE_ASPECT' entities. This corresponds to the path 'no' after the 'Connected to Geometry' check in Fig. 5. These default tolerances apply to all the dimensions for which specific tolerances are not given. Traditionally these tolerances are shown in the 'title block' of the manufacturing drawing. The type and values of default tolerances are extracted from the entities of Part: 1052 (Default tolerances). The values of these default tolerances are then passed on to the tolerance analysis programs.

After extracting the topological and geometric details of the entities, the attached dimension and tolerance information is extracted. The tolerance information is inferred from various entities from the schemas of Part: 47 (Shape variation tolerances). Figure 6 shows the important steps in this process. First, it is determined whether the annotation is a dimension or has a datum or an FCF. If a dimension is attached to the feature, then the type of the dimension, i.e., dimension of size or location or distance or angle, is determined. Then the nominal value is identified. The type and value of tolerance affecting this dimension are determined. The dimensions and tolerance values from the standard entities can be readily extracted from the entities corresponding to Part: 1050 (Dimensions tolerances).

If a datum callout is attached, the name is identified and attached to the part feature. The datum target values are not considered in this paper. In the case of an FCF, all the values related to the geometric tolerances along with the modifiers are extracted as shown in Fig. 6. These values can be extracted from the entity mappings defined in Part: 1051 (Geometric tolerances). Once all the information is extracted, it is linked to the corresponding part feature. The tolerance information and the corresponding feature geometry are further used to define the tolerance zones and perform tolerance analysis, described in the next section.

7 Use-cases

7.1 Assembly use-case

This process is demonstrated using a motor assembly design provided by Mjøs Metallvare shown in Fig. 7. The tolerances applied to these three major motor sub-assemblies are also shown. The tolerance values shown are created for demonstration purposes and do not correspond with the true tolerance values in the product. Figure 8 shows the portion of

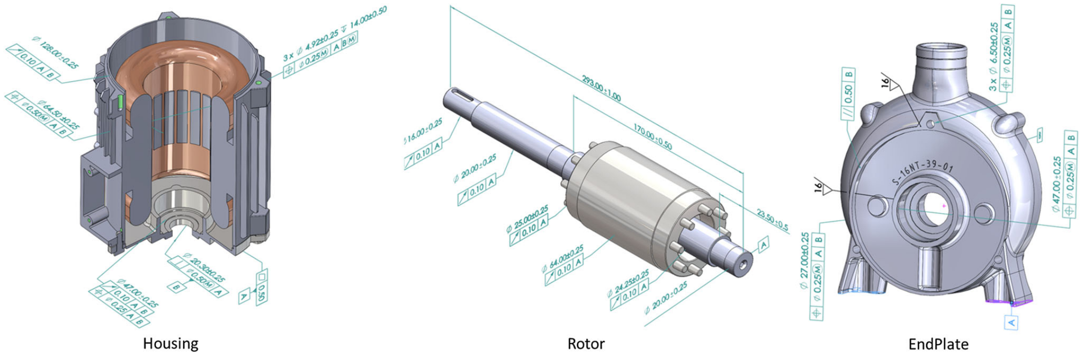
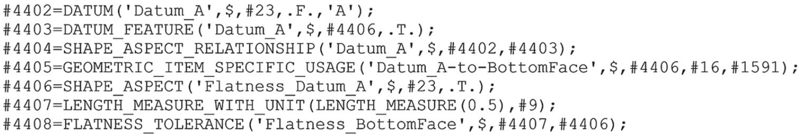


Fig. 7 Motor sub-assemblies with GD&T annotations

Fig. 8 Portion of STEP file showing the entities for semantic tolerance and datum annotations



```
#4402=DATUM('Datum_A',$,#23,.F.,'A');
#4403=DATUM_FEATURE('Datum_A',$,#4406,.T.);
#4404=SHAPE_ASPECT_RELATIONSHIP('Datum_A',$,#4402,#4403);
#4405=GEOMETRIC_ITEM_SPECIFIC_USAGE('Datum_A-to-BottomFace',$,#4406,#16,#1591);
#4406=SHAPE_ASPECT('Flatness_Datum_A',$,#23,.T.);
#4407=LENGTH_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.5),#9);
#4408=FLATNESS_TOLERANCE('Flatness_BottomFace',$,#4407,#4406);
```

a STEP file with standards entities adding a flatness tolerance and a datum callout attached to the FCF of the flatness tolerance. The clearance ratios between the Rotor-Housing and between Rotor-EndPlate are calculated using these tolerance values. The overall process for calculating the effective clearance from the mating features and their tolerances is shown in Fig. 9. The lowest clearance ratio value is considered for programming of the assembly processes where mating/insertion co-occurs at two features. In the present use-case, the insertion/mating happens at two features simultaneously during the assembly of Rotor with Housing. The different stages of this assembly are shown in Fig. 10. The start and end of these two stages of mating can be estimated from the insertion lengths, and the insertion lengths are calculated from the mating constraints and the tolerance values. In this case, the overall insertion length is from the bottom of the bearing seat to its corresponding mating surface on the bearing, approximately 162 mm. The Rotor reaches the Housing in the initial approach and achieves rough alignment. During this stage, the manipulator has much freedom in angle and position. Precise control is unnecessary, and the manipulator path can be planned based on position control.

In the first stage, the first possible contact situation occurs between the magnet region of the Rotor with the windings in the Housing. At the beginning of this stage, the clearance between the components is about 3 mm, and it reduces to 0.25 mm as the insertion proceeds. Precise position control is needed from this stage.

The second stage starts when the bearing starts mating with the bearing seat in the bottom plate of the Housing sub-assembly. This stage starts after insertion of about 142 mm and ends after another 19 mm insertion. The clearance between the housing seat and the bearing outer diameter is around 0.03 mm, requiring sensor-feedback control.

The assembly task is completed when the bearing is seated in the Housing. Successful assembly is achieved when a sufficient force is detected and the Rotor is placed within the expected region identified by tolerance analysis.

Similarly, the assembly of EndPlate with the rest of the motor can be divided into first stage when it starts mating with the shaft and the second stage when the top bearing starts mating with the bearing housing on the EndPlate. In this step, the first stage has very high clearance, and the clearances in the second stage are the same as that of the second stage of the earlier step.

7.2 Welding use-case

The same process is applied to the T–joint of two pipes shown in Fig. 11. In case of structural welds involving beams, the weld seams are along the edges of mating surfaces. These can be directly identified and extracted from the CAD models. Extraction and identification of welds involving cylindrical parts like pipe welds are a bit complicated. Generally welders cut the profiles in the pipes before welding using wrap-around template curves. When these weld joints are properly modelled in the CAD files, then the seam curves will be readily available in the STEP files. The 'B_SPLINE_CURVE_WITH_KNOTS' entity from the STEP AP242 file shown in Fig. 12 gives the weld seam of the T-joint shown in Fig. 11. From this, the nominal geometry of the seam curve was derived. The information from

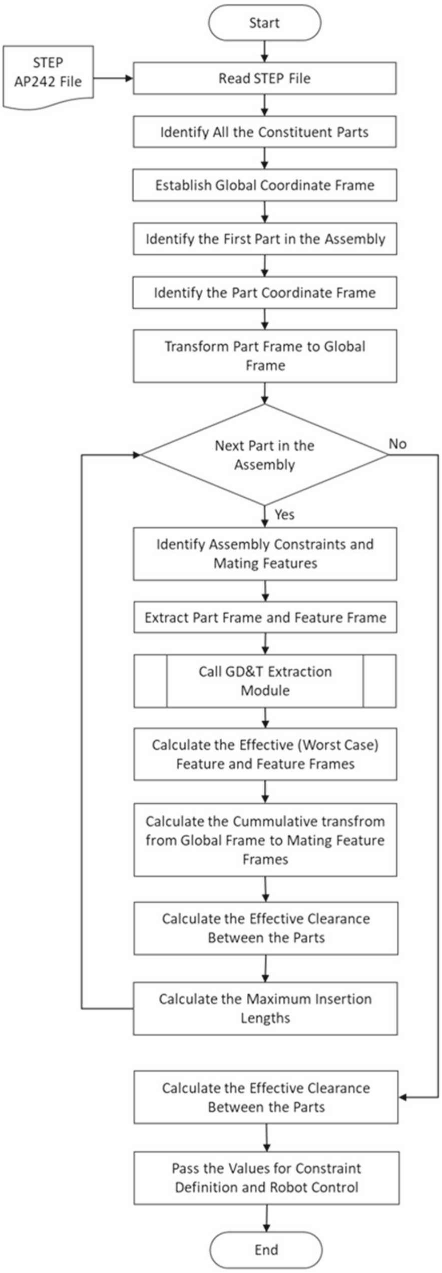

Start
STEP AP242 File
Read STEP File
Identify All the Constituent Parts
Establish Global Coordinate Frame
Identify the First Part in the Assembly
Identify the Part Coordinate Frame
Transform Part Frame to Global Frame
Next Part in the Assembly
No
Yes
Identify Assembly Constraints and Mating Features
Extract Part Frame and Feature Frame
Call GD&T Extraction Module
Calculate the Effective (Worst Case) Feature and Feature Frames
Calculate the Cummulative transfrom from Global Frame to Mating Feature Frames
Calculate the Effective Clearance Between the Parts
Calculate the Maximum Insertion Lengths
Calculate the Effective Clearance Between the Parts
Pass the Values for Constraint Definition and Robot Control
End

Fig. 9 Process for extracting and using assembly and tolerance information

GD&T annotations is used to calculate the deviations from the nominal geometry. The identified weld seams will be in the local coordinate frame of the mating features. These have to be represented with respect to the coordinate frame of the welding cell in which the weld torch is described.

The Unicode string for welding annotation is included as ‘TEXT_LITERAL’ entity. The groove geometry was extracted from the weld annotations. The weld groove is bevel with a bevel angle of 37.5°±2.5° and root face of 1.6 ± 0.4 mm. The Unicode weld string is modified by adding the tolerance value after the nominal value.

- The tolerance value is separated from the nominal value by ‘\v’ as shown in Fig. 12.
- If the tolerance is unilateral the tolerance value has ‘+’ or ‘-’ sign, which ever is applicable. If the tolerance is bilateral the value does not carry any sign.

The groove geometry with the seam can be used as an input for defining the sensing window for planning the scanning using a visual sensor system.

8 Conclusions

This paper described how GD&T information could be included in STEP AP242 files as a neutral exchange file format between in-company processes or automation providers for SMEs. Two methods were described: one used a Unicode string and the other used standard entities from the second edition of the STEP AP242 standard. A process to extract the included GD&T information from STEP files is presented in this paper. The relevance of this GD&T information for robot programming was discussed, and a motor assembly was used as a use-case, demonstrating the use of this information. This GD&T information is used to estimate the worst-case boundaries of the mating features. This information is also used to calculate the clearance ratio and insertion length. These two are essential parameters in an assembly operation that decide the mating forces and can be used to decide the control method for robotic assembly.

The tolerance information can be combined with statistical quality control and machine learning methods to automatically estimate the clearances for the same or similar products in an assembly line. This will increase the effectiveness and applicability of the robotic assembly for batch production.

The Unicode string approach to annotate GD&T is simple and allows for fast parsing and backward compatibility

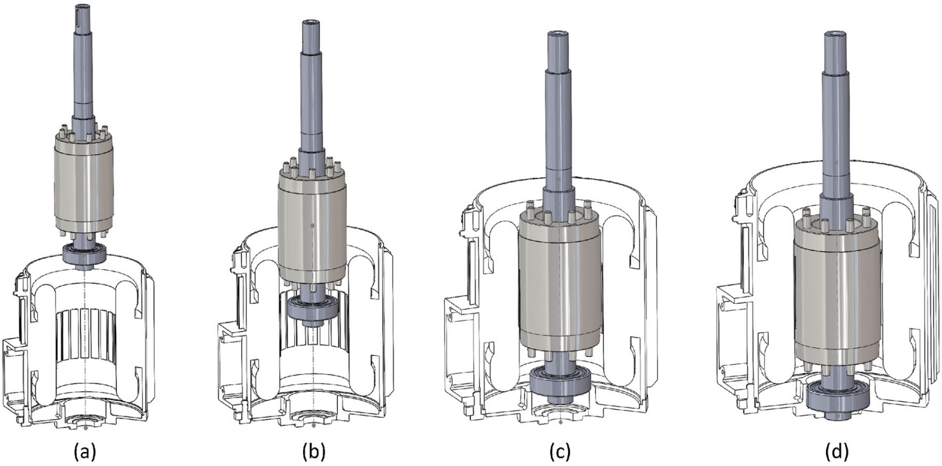


Fig. 10 Rotor—Housing assembly stages. **a** Initial approach. **b** First stage. **c** Second stage. **d** Final stage

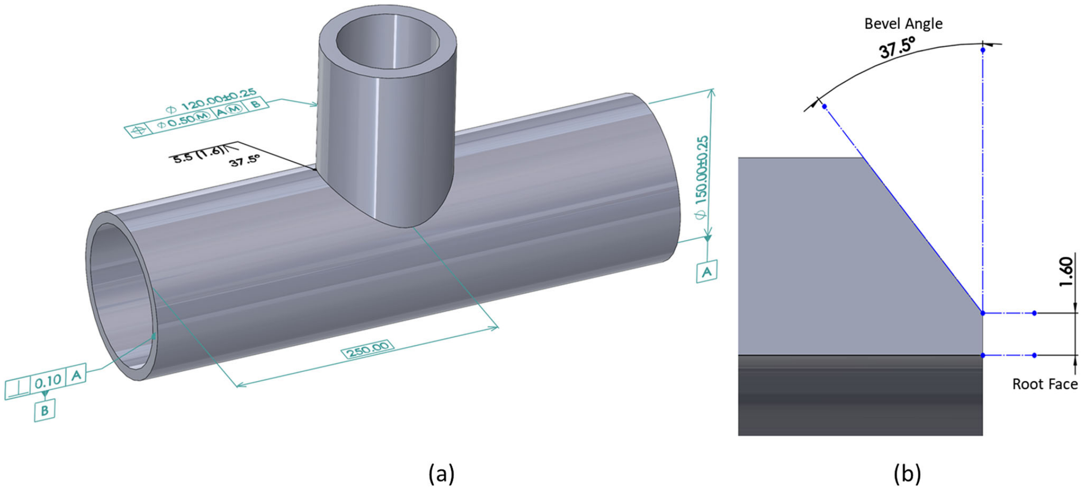


Fig. 11 **a** T-joint of pipes with GD&T and weld annotations. **b** Groove geometry

Fig. 12 Portion of STEP file showing the entities for weld seam and weld annotations

```
#361=B_SPLINE_CURVE_WITH_KNOTS('',3,(#269,#270,#271,#272,#273,#274,#275,#276,
#277,#278,#279,#280,#281,#282,#283,#284,#285,#286,#287,#288,#289,#290,#291,
#292,#293,#294,#295,#296,#297,#298,#299,#300,#301,#302,#303,#304,#305,#306,
#307,#308,#309,#310,#311,#312,#313,#314,#315,#316,#317,#318,#319,#320,#321,
#322,#323,#324,#325,#326,#327,#328,#329,#330,#331,#332,#333,#334,#335,#336,
#337,#338,#339,#340,#341,#342,#343,#344, #345,#346,#347,#348,#349,#350,#351,
#352,#353,#354,#355,#356,#357,#358,#359,#360),.UNSPECIFIED.,.T.,.F.,(4,2,2,2,2,
2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,
2,4),(0., 0.03125,0.046875,0.0625,0.09375,0.125,0.15625,0.1875,0.203125,0.21875,
0.25,0.265625,0.28125,0.3125,0.328125,0.34375,0.375,0.40625,0.4375,0.453125,
0.46875,0.484375,0.5,0.515625,0.53125,0.5625,0.578125,0.59375,0.625,0.640625,
0.65625,0.6875,0.703125,0.71875,0.75,0.765625,0.78125,0.8125,0.84375,0.875,0.90625,
0.9375,0.953125,0.96875,0.984375,1.),.UNSPECIFIED.);
#362=EDGE_CURVE('',#252,#252,#361,.T.);
#439=TEXT_LITERAL('Weld Annotation','WLDNNN\\w\\w\\u\\u\\u\\u\\u\\u\\u\\u\\w5.5
\\u1.6\\v0.4\\uWBAS\\u\\u\\u37.5\\v2.5\\u\\x\\u\\u\\w',$,$,$,$);
```

with previous STEP application protocols. Using the STEP AP242 second edition entities for annotation allows for more granular control and a direct linking between zero, one, or multiple STEP entities. However, it is not available in all CAD software systems yet. In either approach, the availability of GD&T information for use in downstream processes is an essential step towards the generalization of robot programming methods that can help close the gap between design and production.

Acknowledgements The work reported in this paper is partially based on activities within the center for research-based innovation SFI Manufacturing, and partially funded by the Research Council of Norway under contract number 237900. The authors would like to thank Mjøs Metallvare for providing the CAD model for the assembly use-case example.

Funding Open access funding provided by NTNU Norwegian University of Science and Technology (incl St. Olavs Hospital - Trondheim University Hospital)

Declarations

Conflict of interest The authors declare that there is no conflict of interest involved in this research work.

References

1. International Federation of Robotics (IFR): Executive Summary World Robotics 2021 Industrial Robots. International Federation of Robotics (IFR) (2021)
2. International Federation of Robotics (IFR): Executive Summary World Robotics 2020 Industrial Robots. International Federation of Robotics (IFR) (2020)
3. European Innovation Council and SMEs Executive Agency (EISMEA): Advanced Technologies for Industry - General findings: Report on technology trends, technology uptake, investment and skills in advanced technologies. European Commission (2020)
4. Lieberman, L.I., Wesley, M.A.: AUTOPASS: an automatic programming system for computer controlled mechanical assembly. IBM J. Res. Dev. **21**(4), 321–333 (1977)
5. Mosemann, H., Wahl, F.M.: Automatic decomposition of planned assembly sequences into skill primitives. IEEE Trans. Robot. Autom. **17**(5), 709–718 (2001)
6. Thomas, U., Wahl, F.M.: A System for Automatic Planning, Evaluation and Execution of Assembly Sequences for Industrial Robots. In: Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 3, pp. 1458–1464 (2001)
7. Kaufman, S.G., Wilson, R.E.J., Calton, T.L.: The Archimedes 2 Mechanical Assembly Planning System. In: Proceedings of IEEE International Conference on Robotics and Automation, Vol. 4; pp. 3361–3368 (1996)
8. Perzylo, A., Somani, N., Profanter, S., Kessler, I., Rickert, M., Knoll, A.: Intuitive instruction of industrial robots: Semantic process descriptions for small lot production. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2293–2300 (2016)
9. Pane, Y., Arbo, M.H., Aertbelien, E., Decre, W.: A System Architecture for Constraint-Based Robotic Assembly with Sensor-Based Skills. IEEE Trans. Autom. Sci. Eng. (Submitted)
10. Mohammed, S.K., Arbo, M.H., Tingelstad, L.: Leveraging Model Based Definition and STEP AP242 in Task Specification for Robotic Assembly. Procedia CIRP. **97**, 92–97 (2021)
11. Srinivasan, V.: In: An Integrated View of Geometrical Product Specification and Verification. Geometric Product Specification, Verification:Integration of Functionality, pp. 1–11. Springer (2003)
12. Proctor, F., van der Hoorn, G., Lipman, R.: Automating robot planning using product and manufacturing information. Procedia CIRP. **43**, 208–213 (2016)
13. Proctor, F., Franaszek, M., Michaloski, J.: Tolerances and Uncertainty in Robotic Systems. In: Advanced Manufacturing of ASME International Mechanical Engineering Congress and Exposition, Vol. 2 (2017)
14. Biggs, G., McDonald, B.: A survey of robot programming systems. In: Proceedings of the Australasian Conference on Robotics and Automation, pp. 1–3 (2003)
15. Young, K., Pickin, C.G.: Accuracy assessment of the modern industrial robot. Ind. Robot. **27**(6), 427–436 (2000)
16. Greenway, B.: Robot accuracy. Ind. Robot. **27**(4), 257–265 (2000)
17. Slamani, M., Nubiola, A., Bonev, I.: Assessment of the positioning performance of an industrial robot. Ind. Robot. **39**(1), 57–68 (2012)
18. Balanji, H.M., Turgut, A.E., Tunc, L.T.: A Novel Vision-Based Calibration Framework for Industrial Robotic Manipulators. Robot. Comput.-Integr. Manuf. **73**, 102248 (2022)
19. Craig, J.J.: Introduction to Robotics: Mechanics and Control. Pearson Education Inc. (2005)
20. Whitney, D.E.: Quasi-static assembly of compliantly supported rigid parts. J. Dyn. Syst. Meas. Contr. **104**(1), 65–77 (1982)
21. Simunovic, S.: Force information in assembly processes. In: 5th International Symposium on Industrial Robots (1975)
22. Haskiya, W., Maycock, K., Knight, J.: Robotic assembly: chamferless peg-hole assembly. Robotics **17**, 621–634 (1999)
23. Lee, H., Park, S., Jang, K., Kim, S., Park, J.: Contact state estimation for peg-in-hole assembly using gaussian mixture model. IEEE Robot. Autom. Lett. **7**(2), 3349–3356 (2022)
24. Strip, D.R.: Primitives for Robotic Mechanical Assembly: Force Directed Insertions. Robot. Comput.-Integr. Manuf. **6**(4), 283–286 (1989)
25. Bruyninckx, H., Dutre, S., Schutter, J.D.: Peg-on-Hole: A Model Based Solution to Peg and Hole Alignment. In: IEEE International Conference on Robotics and Automation, pp. 1919–1924 (1995)
26. ElMaraghy, H.A., ElMaraghy, W.H., Knoll, L.: Design specification of parts dimensional tolerance for robotic assembly. Comput. Ind. **10**, 47–59 (1988)
27. Chung, S.Y., Lee, D.Y.: Discrete event systems approach to fixtureless peg-in-hole assembly. In: Proceedings of the American Control Conference, pp. 4962–4967 (2001)
28. Takahashi, J., Fukukawa, T., Fukuda, T.: Passive alignment principle for robotic assembly between a ring and shaft with extremely narrow clearance. IEEE/ASME Trans. Mechatron. **21**(1), 196–204 (2016)
29. Dietrich, F., Buchholz, D., Wobbe, F., Sowinski, F., Raatz, A., Schumacher, W., et al: On Contact Models for Assembly Tasks: Experimental Investigation Beyond the Peg-in-Hole Problem on

the Example of Force-Torque Maps. In: The 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2313–2318 (2010)

30. Jin, S., Zhu, X., Wang, C., Tomizuka, M.: Contact Pose Identification for Peg-in-Hole Assembly under Uncertainties. In: 2021 American Control Conference (ACC), pp. 48–53 (2021)
31. Mohammed, S.K., Arbo, M.H., Tingelstad, L.: Constraint Identification from STEP AP242 files for Automated Robotic Welding. In: 2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), pp. 277–282 (2021)
32. Pires, J.N., Loureiro, A., Bolmsjo, G.: Welding Robots: Technology, System Issues and Applications. Springer (2006)
33. Department of Defense, United States of America (USA): MIL-STD-31000A: Technical Data Packages. Federal Standardization Manual. Department of Defense, USA (2009)
34. Herron, J.B.: Re–Use Your CAD: The Model–Based CAD Handbook. CreateSpace Independent Publishing Platform (2013)
35. Urbas, U., Vrabic, R., Vukasinovic, N.: Displaying product manufacturing information in augmented reality for inspection. Procedia CIRP. **81**, 832–837 (2019)
36. Goher, K., Shehab, E., Al-Ashaab, A.: Model-Based Definition and Enterprise: State-of-the-art and future trends. Proc. Inst. Mech. Eng. Part B: J. Eng. Manuf. (2020)
37. ISO 10303–242: Industrial automation systems and integration - Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering. International Organization for Standardization, Geneva, Switzerland (2014)
38. ISO 10303–242: Industrial automation systems and integration - Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering. International Organization for Standardization, Geneva, Switzerland (2020)
39. Kramer, T., Xu, X., 1.: In: Xu, X., Lee, A.Y.C. (Eds.) STEP in a Nutshell. Springer Series in Advanced Manufacturing: Advanced Design and Manufacturing Based on STEP, pp. 1–22. Springer, London (2009)
40. STEP AP242 Project Committee: Development of STEP AP 242 ed2: Managed Model Based 3D Engineering. Whitepaper; STEP AP242 Project Committee. http://www.ap242.org/edition-2 (2014)
41. ASME Y14 41:2019. Digital Product Definition Data Practices. The American Society of Mechanical Engineers. (2019)
42. Praveen, O.V.S., Dileep, B., Gayatri, S., Lawrence, D., Manu, R.: Automated Tolerance Analysis of Mechanical Assembly Using STEP AP242 Managed Model-Based 3D Engineering. Industry 40 and Advanced Manufacturing Lecture Notes in Mechanical Engineering, pp. 149–157 (2021)
43. Feeney, A.B., Frechette, S.P., Srinivasan, V.: A Portrait of an ISO STEP Tolerancing Standard as an Enabler of Smart Manufacturing Systems. ASME J. Comput. Inf. Sci. Eng. **15**(2), 021001–021001–5 (2015)
44. Zhong, R.Y., Xu, X., Klotz, E.: Newman ST. Intelligent Manufacturing in the Context of Industry 4.0: A Review. Eng. **3**(5), 616–630 (2017)
45. Thomas, R.G., Lawrence, D., Manu, R.: STEP AP242 Managed Model-based 3D Engineering: An Application Towards the Automation of Fixture Planning. Int. J. Autom. Comput. **18**(5), 731–746 (2021)
46. ISO 10303-21: Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure. International Organization for Standardization, Geneva, Switzerland; (2016)
47. CAx Implementor Forum: CAx-IF Recommended Practices for the Representation and Presentation of Product Manufacturing Information (PMI) (AP242). Version 4.0 (2014)
48. ASME Y14 5: Dimensioning and Tolerancing. The American Society of Mechanical Engineers; (2018)
49. ISO 1101: Geometrical product specifications (GPS) - Geometrical tolerancing—Tolerances of form, orientation, location and run-out. International Organization for Standardization, Geneva, Switzerland (2017)
50. Lipman, R.R., Filliben, J.J.: Testing implementations of geometric dimensioning and tolerancing in CAD software. Comput-Aided Des. Appl. **17**(6), 1241–1265 (2020)
51. AFNeT. Test Report for the STEP AP242 Benchmark #3: CAD Test Cases - Short Report. 30 Rue de Miromesnil, 75008 Paris: AFNeT (2020)
52. AFNeT.: Test Report for the STEP AP242 Benchmark #2: CAD Test Cases - Short Report v1.1. 30 Rue de Miromesnil, 75008 Paris: AFNeT (2017)
53. CAx Implementor Forum: PMI Unicode String Specification Examples and Mapping Strategies - for Dimensioning and Tolerancing, GD&T, Surface Texture Symbol, and Welding Symbol PMI Annotation Entities. Rev. J. (2011)
54. LKSoftWare GmbH, Germany: JSDAI. https://www.jsdai.net
55. ISO 10303-22: Industrial automation systems and integration - Product data representation and exchange - Part 242: Implementation methods: Standard data access interface. International Organization for Standardization, Geneva, Switzerland (1998)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

A.4. Paper IV:

Shafi K. Mohammed, Mathias H. Arbo, Lars Tingelstad, “Constraint Definition for Gripper Selection and Grasp Planning for Robotic Assembly using Product Manufacturing Information from STEP AP242Ed2 Files,” (Machines 2022)

machines

Article

Constraint Definition for Gripper Selection and Grasp Planning for Robotic Assembly Using Product Manufacturing Information from STEP AP242Ed2 Files

Shafi Khurieshi Mohammed [1],*, Mathias Hauan Arbo [2] and Lars Tingelstad [1]

[1] Department of Mechanical and Industrial Engineering, Norwegian University of Science and Technology (NTNU), 7034 Trondheim, Norway
[2] Research Scientist, SINTEF Manufacturing AS, 7031 Trondheim, Norway
* Correspondence: shafi.k.mohammed@ntnu.no;

Abstract: This article uses the Product Manufacturing Information (PMI) from STEP AP242 neutral files for gripper selection and grasp planning in a robotic assembly operation. The PMI, along with the part geometry and dimensions, are used in identifying various handling features of the parts and selecting an appropriate gripper. The required PMI, like material, volume, surface finish, threading and coating information, are added to the STEP AP242 files. The PMI is semantically included in the STEP files following the Model Based Definition (MBD) methodology. Two methods are described to add the PMI to the STEP files, one using a custom string and another using the standard entities defined in ISO 10303 AP242: 2020 standard. The entire process is demonstrated in a use case.

Keywords: robot; assembly; STEP AP242; ISO 10303; STEP; Model Based Definition (MBD); Product Manufacturing Information (PMI); constraint based robot programming

Citation: Mohammed, S.K.; Arbo, M.H.; Tingelstad, L. Constraint Definition for Gripper Selection and Grasp Planning for Robotic Assembly Using Product Manufacturing Information from STEP AP242Ed2 Files. *Machines* **2022**, *10*, 1230. https://doi.org/10.3390/machines10121230

Academic Editors: Yuansong Qiao and Seamus Gordon

Received: 12 September 2022
Accepted: 13 December 2022
Published: 16 December 2022

1. Introduction

Industrial grippers are crucial in many types of equipment like NC and special purpose machines, fixed automation, workpiece turrets and industrial manipulators [1,2]. The problem of robotic grippers is not new [3], but with the increasing adoption of robotic applications under Industry 4.0 [4], their importance is increased as they are the essential tool in many industrial operations like material handling [2]. Grippers are an essential component for the success of robotic assembly operation [5] as they affect both the cost and time of automation [6]. Hence, selecting an appropriate gripper and proper grasp planning will increase the successful completion of assembly operations with less time and lower cost.

The gripper selection and grasp planning is critical in the case of small and medium enterprises (SMEs) as they handle significant product variations and lower production volumes compared to mass production. As the product design changes, the SME needs to make the gripper selection and grasp planning for each change, which results in increased time and cost required for robotic assembly. The gripper selection and grasp planning process can be improved using product information. Product information is made available for all the stakeholders and downstream operations by adopting the Digital Thread (DT) methodologies as per Industry 4.0.

Mohammed et al. [7] used a method to add welding information semantically to the STEP files. This paper extends that method to include relevant PMI to the STEP AP242 Ed2 files and reuse it to identify the constraints for grasp planning. We also identify the information required for gripper selection and grasp planning from the existing literature and how this information can be included in the STEP AP242 files as per the latest industry standards. A method to extract the relevant information from the STEP files and identify the constraints for gripper selection and grasp planning is proposed. This process is demonstrated using a motor assembly use case.

The paper is structured as follows. Section 2 describes the gripping process and information needed for gripper selection and grasp planning from the existing literature. Section 3 presents the methodology of Model Based Definition (MBD) and the STEP AP242 standard is described in Section 4. Sections 5–7 demonstrate how surface finish and other PMI can be added to the STEP AP242 files. Section 8 describes the process of extracting the constraint information from the STEP AP242 files and the process is demonstrated using a use case. The concluding remarks are presented in Section 9.

2. Robotic Gripping

Successful adoption of automatic assembly systems depends on the capabilities of the grippers [1]. The gripping problem is one of the critical issues in the industrial assembly and part handling process. The grasp planning processes are classified into analytical and data-driven methods [8,9] as shown in Figure 1. Analytical methods are based on kinematic and dynamic models of grasp problem and the data-driven methods are based on machine learning techniques. Carvalho de Souza et al. [10] classified the data-driven methods by separating the deep learning methods from the rest of the machine learning methods. Kleeberger et al. [11] presented a comparison of various vision-based learning methods of grasp planning for handling rigid, articulated and deformable objects.

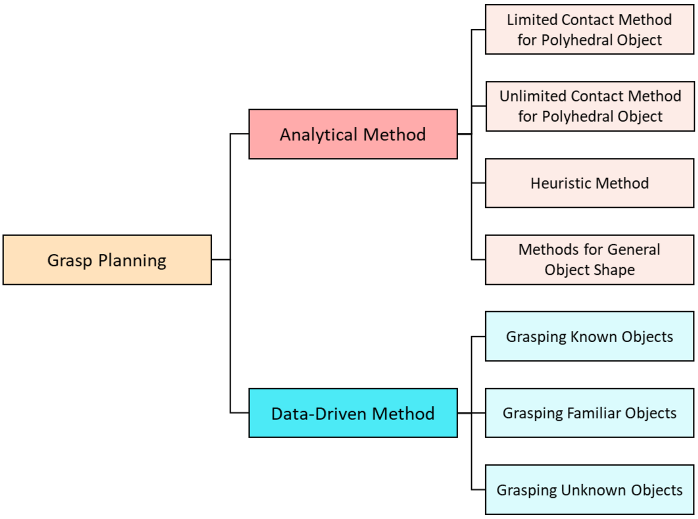

Grasp Planning
Analytical Method
Data-Driven Method
Limited Contact Method for Polyhedral Object
Unlimited Contact Method for Polyhedral Object
Heuristic Method
Methods for General Object Shape
Grasping Known Objects
Grasping Familiar Objects
Grasping Unknown Objects

Figure 1. Classification of grasp planning [8,9].

2.1. Gripper Performance and Selection

The grasp planning and gripper selection depend on several factors, but most are related to the objects being handled. As the gripper directly comes in contact with the parts during handling and assembly, their function and working envelope depend on the object's properties. The parameters that influence the selection and functioning of grippers are well established in the literature [5,12–14].

As per Owen [12], the function of the gripper is limited by

- Static force;
- Dynamic force;
- Part geometry.

Birglen et al. [15] presented a comparison of the two-finger parallel industrial grippers from various gripper manufacturers. They used characteristics like force, stroke, gripper weight and C-factor for comparing the different out-of-shelf grippers. C-factor is a measure that enables the comparison of grippers of different sizes and designs. It is calculated by multiplying the gripper's stroke length by the ratio between the force and weight [2]. The object size, shape and material properties determine the selection of a gripper with appropriate force, stroke and C-factor.

Lotter [5] divided the object parameters that affect the handling tasks into characteristics and behavior. Object characteristics included geometry, size, mating/locking features, holes and physical properties such as material and temperature. Behavior parameters are further divided into stationary behavior that include positional stability and transfer behavior. These parameters are derived for both fixed and robotic automation. Some of these parameters are not relevant for robotic gripping, like stackability.

Fantoni et al. [14] identified the relevant parameters for gripper selection from the Design for Assembly principles. These parameters were grouped as object and operation parameters. The object parameters included both physical and geometric properties of the object and the operation parameters included the factors related to feeding, handling and placing.

2.2. Grasp Planning Using CAD

The closure of grasp in analytical methods is generally classified into two types depending on the method of constraints.

Force-closure Grasps The grasp is known as force-closure if the object is held in position by the gripper by applying force and moment at the contact points [16].

Form-closure Grasps Form-closure is achieved by selecting the contact points such that the object is constrained without applying any frictional forces or moments [17,18].

Other forms of grasps like partial form-closure are also available in the literature [18].

The analytical methods of grasp planning use the part shape, features, material and surface conditions in selecting the contact points and estimating the forces to achieve a successful grasp. The data-driven methods also depend on the earlier knowledge of the object's shape or similarity with a known shape. When the object shape is entirely unknown, the data-driven methods use sensor data to understand the object shape [8]. Even the data-driven methods benefit from the previous knowledge of the object geometry and increase the grasp success [10].

Analytical methods are more appropriate for industrial applications than the data-driven methods [9]. The grasp planning can benefit from using the CAD models of the parts being handled and assembled. Kleeberger et al. [19] demonstrated that the success rate of data-driven methods is increased by using object geometry either from CAD models or point clouds. Many researchers used part geometry for grasp planning [20]. van Bruggen [21] described a method for identifying constraints of part features/surfaces unsuitable for gripping. Miller et al. [22] used primitive shapes like spheres, cylinders, cones and boxes for identifying initial grasp points and the grasp is simulated to arrive at the best option.

Schmalz et al. [23] presented a method for dimensioning grippers using part geometry from STEP files. Somani [24] combined CAD data with vision sensors for pose estimation and defining constraints for bin-picking tasks.

This paper considers only the physical and geometric object properties and how these parameters can be retrieved from the part design data for robotic applications. This paper also uses assembly constraints and threading information for defining constraints for grasp planning. The following key object parameters [1,14,25]:

- Dimensional size
- Material: strength, hardness, density
- Weight
- Surface texture
- Special coatings
- Thread information [21]
- Assembly constraints and mating information

- Center of gravity
- Shape
- Part features: holes, surfaces

3. Model-Based Definition (MBD)

The 3D CAD models are prepared by the designers during the design phase of the product life cycle. These 3D models are used to prepare the 2D manufacturing drawings. The manufacturing drawings have the complete product definition in terms of 2D geometry, dimensions and other PMI. Hence, the 2D manufacturing drawings are the single source of product definition and are used as input to all other downstream operations in the product life cycle. Manual effort is needed to read and recreate the operation-specific information from the manufacturing drawings at each downstream operation. The reinterpretation and recreation of data on the manufacturing shopfloor may result in human errors leading to part quality issues that, in turn, result in increased time-to-market and production costs. The manual intervention can be avoided by directly using the product data in the downstream operations. The direct reuse of product definition is facilitated by adopting a Model-Based Definition (MBD) methodology.

In the MBD methodology, the PMI is directly added to the features of the 3D CAD model during the design stage rather than preparing a separate 2D manufacturing drawing. Hence, the 3D model becomes the master source of product information carrying the product geometry, critical dimensions, GD&T, surface finish and all other information required by the downstream operations [26,27]. Figure 2 shows a 3D CAD model of a motor end plate with all the PMI attached to its features along with title-block information like part number, material and release date. MBD is the building block of product information that enables the establishment of DT in a connected Model Based Enterprise (MBE). Many downstream operations like NC machining and automated inspection [28] can reuse the product information directly from the MBD without manual intervention. Goher et al. [29] recorded many state-of-the-art applications of MBD.

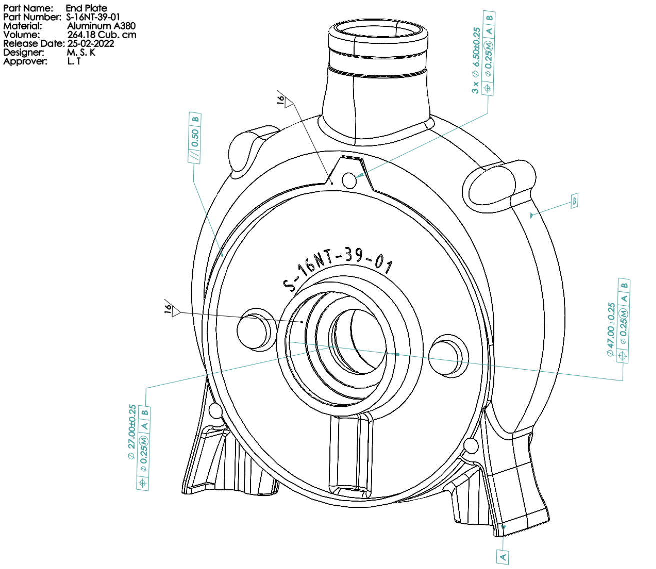


Figure 2. Part with Material, Volume, Surface Texture and GD&T Annotations.

The product information can be added to the 3D model as presentation PMI or semantic PMI. The presentation PMI serves the purpose of visualizing PMI on a 3D model. This type of PMI defeats the purpose of MBD as it requires human intervention to read and reinterpret the product data from the 3D model. Semantic PMI can be read and interpreted using computer programs. It serves the purpose of MBD by realizing the goal of reducing or eliminating human involvement in data recreation at the downstream manufacturing operations.

Another aspect related to the product definition is the problem of sharing it with downstream operations and other stakeholders like sub-contractors. The designers can share the MBD with the manufacturing teams in the native CAD format in which they were created. However, in this case, all the stakeholders require access and expertise to use the same commercial CAD software. Many SMEs cannot afford to purchase the licenses of the CAD software used by the original equipment manufacturers (OEMs) to whom they supply. Hence, the need for a neutral exchange format that can be used to share product definitions to all the stakeholders across the extended enterprise. STEP AP242 neutral file format matches the requirements for sharing the MBD as it can capture the PMI along with the product geometry. A brief description of STEP AP242 is provided in the following section.

4. STEP AP242

The Ssandard for the Exchange of Product Model Data (STEP) is the standard for product data representation and exchange extensively used in the industry. STEP AP242 is the latest member of the STEP-family of ISO (International Standards Organization) standards *10303: Automation systems and integration–Product data representation and exchange*. The application protocol *AP242: Managed model–based 3D engineering* is defined to fulfill the needs of capturing and exchanging MBD [30]. The first edition of AP242, released in 2014, replaced the earlier application protocols AP203 (configuration controlled 3D designs of mechanical parts and assemblies) and AP214 (core data for automotive mechanical design processes) [31,32]. The latest (second) edition of this Application Protocol (AP) was in 2020 [33]. With the increasing digitalization as part of Industry 4.0 processes, the significance and use of STEP AP242 are increasing. STEP AP242 is used for various applications like automated tolerance analysis [34], CAD/CAE integration and smart manufacturing [35–37].

STEP AP242 facilitates the creation of MBD as per *ASME Y14.41–2019: Digital Product Definition Data Practices* [38] standard. Both presentation and semantic PMI can be added to the STEP AP242 files. Only semantic PMI is used in the current paper as presentation PMI is not suitable for robot programming and constraint definition. STEP AP242 files can carry the following product information.

Geometry STEP files capture the exact product geometry that can be extracted and used to identify various features of the part.

Assembly Information The information about the product structure and assembly constraints are part of STEP AP242 standards. This information is used to identify the relative positions of components in an assembly and is used by Mohammed et al. [39] to define the robot motion constraints for assembly.

Critical Dimensions and GD&T The critical-to-quality dimensions and the GD&T information can be directly added to the relevant part features in a STEP AP242 file. The information can be used to extract the sizes and locations of assembly features and estimate possible deviations from the nominal design values as described by Mohammed et al. [40].

Annotations STEP AP242 enables the addition of other necessary PMI such as surface finish, thread specifications and welding information as annotations. Mohammed et al. [7] added the welding annotations to the weld seams using Unicode strings to the STEP AP242 files.

Properties The product properties like material, weight, bounding box and other metadata can be included in the STEP AP242 files and reused in the automation of handling and packaging.

Notes Text annotations can be added to the STEP AP242 files to include tool or process-related information needed in the downstream manufacturing operations.

This paper deals with the addition and extraction of surface texture, thread, coating, bounding box and material and part properties to the STEP AP242 files.

5. Surface Finish Symbols

The addition of surface finish symbols in the product documentation should be as per *ISO 21920-1:2021-Geometrical product specifications (GPS)—Surface texture: Profile—Part 1: Indication of surface texture* [41]. *ISO 21920-1: 2021* replaced the earlier standard *ISO 1302:2002-Geometrical Product Specifications (GPS)—Indication of surface texture in technical product documentation* [42]. These ISO standards deal with the symbols used in the profile methods. *ISO 25178-1:2016-Geometrical product specifications (GPS)—Surface texture: Areal—Part 1: Indication of surface texture* [43] specifies the rules for adding areal surface texture symbols. Table 1 shows the surface texture symbols from these standards.

Table 1. Various profile and areal surface texture symbols and the corresponding standards [41–43].

Symbol	Type	Name	Description	ISO Standard
	Profile	Basic symbol	Any manufacturing process permitted	1302:2002 (Obsolete)
	Profile	Expanded symbol	Material removal required	1302:2002 (Obsolete)
	Profile	Expanded symbol	Material removal not permitted	1302:2002 (Obsolete)
	Profile	Complete symbol	Any manufacturing process permitted	1302:2002 (Obsolete)
	Profile	Complete symbol	Material removal required	1302:2002 (Obsolete)
	Profile	Complete symbol	Material removal not permitted	1302:2002 (Obsolete)
	Profile	Graphical symbol	Any manufacturing process permitted	21920-1:2021
	Profile	Graphical symbol	Material removal	21920-1:2021
	Profile	Graphical symbol	Material removal not permitted	21920-1:2021
	Areal	Basic symbol	Any manufacturing process permitted	25178-1:2016
	Areal	Expanded symbol	Material removal required	25178-1:2016
	Areal	Expanded symbol	Material removal not permitted	25178-1:2016
	Areal	Complete symbol	Any manufacturing process permitted	25178-1:2016
	Areal	Complete symbol	Material removal required	25178-1:2016
	Areal	Complete symbol	Material removal not permitted	25178-1:2016
	Areal	Complete symbol	Showing ‘All Around’ symbol	25178-1:2016

ISO 21920-1:2021 standard defines two levels of parameter indication on the surface finish symbols. One is the minimal level of indication, where only mandatory parameters or mandatory parameters along with one of the semi-mandatory parameters are shown on the symbol and all other parameters are mapped to their default values. Another complete level of indication specifies values of all the parameters on the surface finish symbol. The complete indication of all the parameters is not common in practice and is used in rare cases [41]. This paper combines both these methods and uses it to form a single custom text string.

Figure 3 shows the surface finish symbol and positional labels of all the possible parameters that can be included in the symbol as per the ISO 21920-1:2021 standard [41].

The parameters shown in between angular brackets ('<' and '>') are mandatory while those shown within square brackets ('[' and ']') are optional when default values are used. The parameters shown between parentheses ('(' and ')') are semi-mandatory. The profile L-filter nesting index determines the default values of all optional parameters in the case of the R-parameter symbol and the profile S-filter nesting index determines the default values of all optional parameters in the case of the W-parameter symbol. Only the relevant (profile L-filter/S-filter nesting index) of these semi-mandatory parameters are shown with the mandatory parameters. When only the two mandatory parameters are shown, then these two semi-mandatory parameters will be considered optional. The positional labels and the description of the parameters represented by these labels shown in Figure 3 are given below [41].

a: tolerance type
b: symbol for R-parameter/P-parameter/W-parameter
c: tolerance limit value of the profile surface texture parameter
d: tolerance acceptance rule
e: profile S-filter type
f: profile S-filter nesting index
g: profile L-filter type
h: profile L-filter nesting index
i: section length
k: number of sections
m: evaluation length, used when indicating evaluation length replaces the section parameters i and k
n: profile F-operator association method and element
p: profile F-operator nesting index
q: method for profile extraction
r: other requirements using OR(n) symbol
s: manufacturing process
t: surface lay and direction of lay
u: profile direction relative to the surface lay

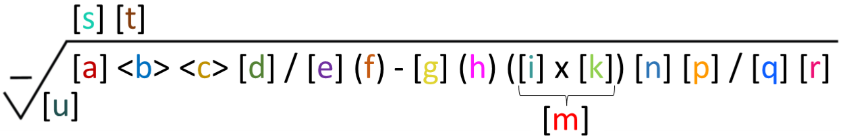

[s] [t]
[a] <b> <c> [d] / [e] (f) - [g] (h) ([i] x [k]) [n] [p] / [q] [r]
[u]
[m]

Figure 3. Format of the surface texture symbol with all the possible parameters as per the ISO 21920-1:2021 standard [41].

The detailed description of these parameters and their possible values are well known in the industry and the discussion about them is considered out of the scope of this paper. The interested reader can refer the relevant standards in the ISO 21920 [44,45] and ISO 25178 [46,47] series. The symbolic representations of the values of some of these parameters are shown in Table 2.

Table 2. Surface texture parameters, the positional labels, symbolic representation and their description [41].

Symbol	**Label**	**Parameter**	**Description**	**Custom Code**
Tmax	d	Tolerance acceptance rule	Maximum tolerance	STMX
T16%	d	Tolerance acceptance rule	16% tolerance	ST16
Tmed	d	Tolerance acceptance rule	Median tolerance	STMD
M	t	Surface lay	Multi-directional	SMLD
C	t	Surface lay	Circular	SCRC
R	t	Surface lay	Radial	SRDL
P	t	Surface lay	Particulate	SPRT
=	t	Surface lay and direction of lay	Parallel	SPRL
⊥	t	Surface lay and direction of lay	Perpendicular	SPPD
X	t	Surface lay and direction of lay	Crossed	SCRS
‡	u	Profile direction	Perpendicular to the predominant direction	SPPD
⥱	u	Profile direction	Parallel to the predominant direction	SPRL
◎	u	Profile direction	Circular	SCRC
≠	u	Profile direction	At an angle to the predominant direction	SA<XX>

6. Adding Surface Finish Symbols to the STEP AP242 Files

This section describes two methods of adding surface finish information to STEP AP242 files. The first method uses a custom-defined text string and the second uses the standard entities defined in the second edition of the STEP AP242 standard.

6.1. Using Custom Annotation String

Mohammed et al. [7,40] demonstrated the addition of welding annotations and GD&T to the STEP AP242 files. It was based on the CAx recommendation of *PMI Unicode String Specification Examples and Mapping Strategies* [48]. The current method is also based on this recommendation and extends the method presented in the earlier papers by Mohammed et al. [7,40]. The differences of the present method from the earlier ones are

- The method described in the CAx recommendation [48] is based on ISO 1302: 2002. However, the present method is extended to match the latest ISO standard 21920-1: 2021, which replaced the earlier standard ISO 1302: 2002.
- The current method deals with the profile method of surface texture annotations. It incorporates the features for adding surface texture annotations as per the areal method specified in ISO 25178-1: 2016.
- While Unicode characters are used for welding and GD&T annotations, this method does not use any Unicode characters. The surface texture annotation has only a few symbols which can be directly mapped to the Unicode characters and most of the values used in surface finish symbols are text-based. Hence, custom codes are used for all the parameters in the surface texture annotation.

Using the current method, the surface texture annotation is fully semantic. For the grasp planning and gripper selection, the semantic interpretation of all the parameters of the annotation are not needed, but this can be useful in automatic quality planning and inspection processes.

Forming the Annotation String:

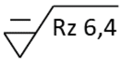
The custom annotation string is formed as per the following rules [7,40,48].

- The annotation string starts with a six-letter combination 'STSXXX'. The first three letters in this combination 'STS' stands for 'Surface Texture Symbol'.
- The following letter is used to indicate the methodology of surface texture symbols whether profile or areal. The next two letters are used to specify the symbol type and manufacturing/machining, respectively. The letter identifiers for these parameters are shown in Table 3.
- The 'All Around' symbol is added after this six-letter combination using 'SAAS' using '\w' as a separator between the two combinations.
- '\w' is used as a separator between various parameters and regions. '\u' is used to separate different values/combinations under a single parameter, for example, separating section length and number of sections. Therefore, the string values representing the section length (i) and number of sections (k) will be of the form '\w*i*\u*k*'. When the evaluation length (m) is given instead of section length (i) and number of sections (k), then the string equivalent will be '\w*m*\u'.
- A four-letter combination such as 'SXXX' is used to indicate various abbreviations, symbols and identifiers. Table 2 shows the various custom codes for some of the symbols used in surface texture annotation.
- The profile direction value 'at an angle' can be included using 'SA<XX>'. Here, the angle value is added in the place of '<XX>', for example 'SA45'.
- The direction of lay and profile direction may carry a 'Feature Control Frame' to indicate the direction and the datum plane. This can be added using '\u' after the direction of lay/profile direction symbol and the datum indicator is separated from the direction in the feature control frame using '\x'.
- Any other values inside various parameters can be added using the separator '\v'. For example, the R-parameter values 'Rdc(*p*, *q*)' can be added as '\wRdc\v*p*\v*q*'.

Table 3. Symbol descriptions and their alphabet identifiers.

Description	Identifier
Profile	P
Areal	A
Basic symbol	B
Expanded symbol	E
Complete symbol	C
Graphical symbol	G
Any manufacturing process permitted	A
Material removal required	M
Material removal not permitted	N

Figure 4 shows the surface finish symbol with mandatory values of R-parameter and tolerance limit along with the equivalent annotation string. Here, all the other parameters are to be mapped to the default values as per ISO 21920-3:2021 [45]. This surface texture string can be customized to match the industry and organizational practices. The method described here is focused on the profile surface texture symbols as prescribed in ISO 21920-1: 2021 [41]. However, the areal surface texture symbol is also enabled in the starting six-letter combination and the rest of the string can be easily formed by adopting the rules for areal symbols.



STSPGM\w\wRz\w6,4\w \w \w \w \w \w \w \w \w \w \w \w \w \w

Figure 4. Surface texture symbol with R-parameter and tolerance limit value and the corresponding annotation string.

The surface texture annotation string can be added to the STEP AP242 files using the entity ‘TEXT_LITERAL’. This annotation can be semantically attached to the relevant part feature using ‘SHAPE_REPRESENTATION’ entity.

6.2. Using the Standard Entities from STEP AP242Ed2

The latest edition of STEP AP242 standard has the capability to add the surface texture symbols as per ISO 1302:2002 and ASME Y14.36M-1996 standards. In the latest edition of the standard, the information requirements of Application Module *Surface conditions* (Part: 1110) are fully developed to match the needs of the surface texture symbols. The following type and entity definitions are given in the standard to include surface texture information in the STEP files.

- Type Definitions:
  - surface_lay_and_orientation
  - surface_texture_characteristic_type
  - surface_texture_material_removal_condition_enumeration
  - surface_texture_requirement_type
- Entity Definitions:
  - Standard_surface_texture_parameter
  - Surface_texture
  - Surface_texture_parameter
  - User_defined_surface_texture_parameter

Though these entities are defined in line with the now-withdrawn ISO 1302:2002 standard, they can be used to represent most of the parameters under the latest surface texture standard ISO 21920-1:2021. Table 4 indicates the matching between the parameters of the surface texture symbol of Figure 3 to the standard entities and attributes.

Table 4. Mapping of surface texture symbol parameters to STEP entities and attributes.

Entity.Attribute	**Indicator**	**Parameter**
Standard_surface_texture_parameter.tolerance_type	a	tolerance type
Standard_surface_texture_parameter.characteristic_type	b	symbol for R-parameter/P-parameter/W-parameter
Standard_surface_texture_parameter.characteristic_value	c	tolerance limit value of the profile surface texture parameter
–	d	tolerance acceptance rule
Standard_surface_texture_parameter.transmission_band_filter_short_wave	e	profile S-filter type
–	f	profile S-filter nesting index
Standard_surface_texture_parameter.transmission_band_filter_long_wave	g	profile L-filter type
–	h	profile L-filter nesting index
Standard_surface_texture_parameter.evaluation_length	i	section length
Standard_surface_texture_parameter.number_of_sampling_lengths	k	number of sections
Standard_surface_texture_parameter.evaluation_length	m	evaluation length
–	n	profile F-operator association method and element
–	p	profile F-operator nesting index
–	q	method for profile extraction
Standard_surface_texture_parameter.additional_information	r	other requirements using OR(n) symbol
Surface_texture.manufacturing_method	s	manufacturing process
Surface_texture.direction	t	surface lay and direction of lay
–	u	profile direction relative to the surface lay

These entities and attributes are mapped to the entities of Integrated Generic Resources: *Fundamentals of product description and support* (Part: 41) and *Material and other engineering properties* (Part: 45). The entity ‘STANDARD_SURFACE_TEXTURE_PARAMETER’ maps to ‘PROPERTY_DEFINITION’ entity of *Product property definition schema* of Part: 41. The entities ‘DESCRIPTIVE_REPRESENTATION_ITEM’ and ‘MEASURE_REPRESENTATION_ITEM’ from *Qualified measure schema* of Part: 45 can be used to add the values to the descriptive and physical measurable attributes, respectively. Figure 5 shows the method of adding the parameter ‘tolerance acceptance rule’, which cannot be matched directly to any entity or attribute in the standard.

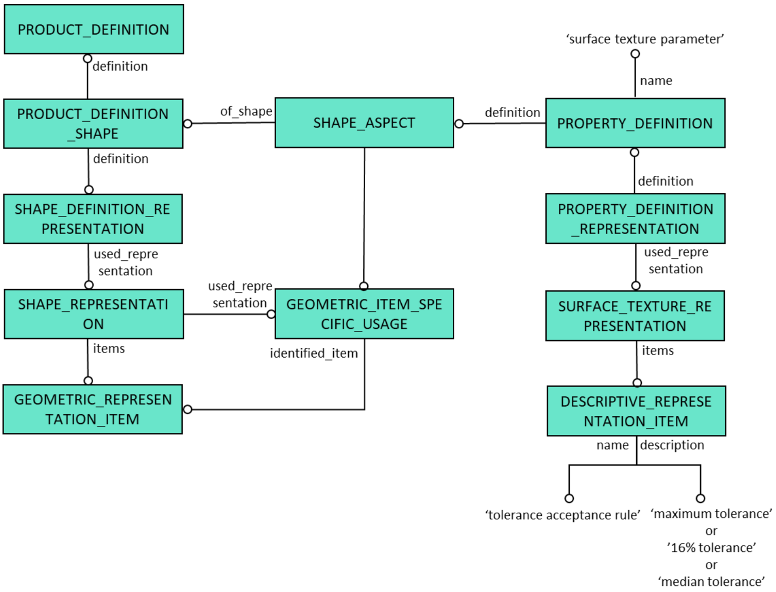

PRODUCT_DEFINITION
definition
PRODUCT_DEFINITION_SHAPE
of_shape
SHAPE_ASPECT
definition
PROPERTY_DEFINITION
'surface texture parameter'
name
definition
SHAPE_DEFINITION_REPRESENTATION
definition
PROPERTY_DEFINITION_REPRESENTATION
used_representation
used_representation
SHAPE_REPRESENTATION
used_representation
GEOMETRIC_ITEM_SPECIFIC_USAGE
SURFACE_TEXTURE_REPRESENTATION
items
identified_item
items
GEOMETRIC_REPRESENTATION_ITEM
DESCRIPTIVE_REPRESENTATION_ITEM
name
description
'tolerance acceptance rule'
'maximum tolerance' or '16% tolerance' or 'median tolerance'

Figure 5. Addition of surface texture parameter—tolerance acceptance rule to STEP AP242 file.

Addition of Custom Annotation String Using Standard Entities

It was suggested in Section 6.1 that the custom string of surface texture annotation can be added using the entity 'TEXT_LITERAL.' The custom annotation string can also be added using the method depicted in Figure 5. The custom string should be added to the 'description' attribute of the 'DESCRIPTIVE_REPRESENTATION_ITEM' entity and 'user defined string representation' should be used for the 'name.'

7. Adding Other Relevant PMI to STEP Files

7.1. Thread Information

Thread specification is one of the key information items about the joining process during the assembly. Generally, designers do not model the exact geometry of the thread features. The thread is specified using 'cosmetic thread', a feature provided by CAD tools. In the manufacturing drawing the threaded region is annotated with a thread note. The same thread annotation can be added to the corresponding geometric element in the STEP file using the method described in Section 6.1. The thread annotation will start with the letter combination 'THD' and other four-letter codes will start with 'TD'. The custom string can be formed to match the needs of metric or unified thread specifications.

7.2. Special Coatings and Paints

The information about the special coatings and paints on part surfaces can be added as specified in the Application Module *Surface conditions* (Part: 1110). The entity 'COATING_LAYER' defines the properties of coatings. This entity is mapped to the entity 'SHAPE_ASPECT' from the Integrated Generic Resource: *Fundamentals of product description and support* (Part: 41).

The coating material can be specified using the entity 'MATERIAL_IDENTIFICATION' of the Application Module *Generic Material Aspects* (Part: 1681) which maps to the entity 'MATERIAL_DESIGNATION' of the Integrated Generic Resource: *Material and other engineering properties* (Part: 45).

7.3. Material

The Application Module *Generic Material Aspects* (Part: 1681) specifies the information requirements for adding material and material properties like density and maps these requirements to the Integrated Generic Resource: *Material and other engineering properties* (Part: 45). Using the entity 'MATERIAL_DESIGNATION' from *Material property definition schema*, material can be added to the STEP files at part level as shown in Figure 6.

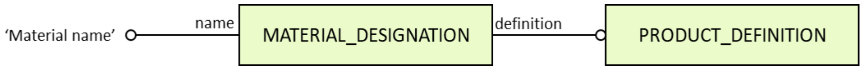


Figure 6. Assignment of a material to a part.

The material specification here is added at the part level by directly referencing the 'PRODUCT_DEFINITION' entity, while the material for coatings (Section 7.2) is defined for the particular surface of the part using 'SHAPE_ELEMENT' entity.

Material Properties

Any material property like density, hardness and strength can be added to the STEP files using the other entities defined in the same Integrated Generic Resource: *Material and other engineering properties* (Part: 45). Figure 7 shows the method of adding the density at the part level.

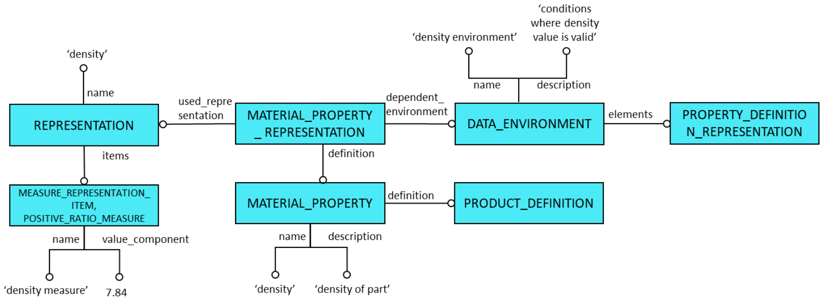


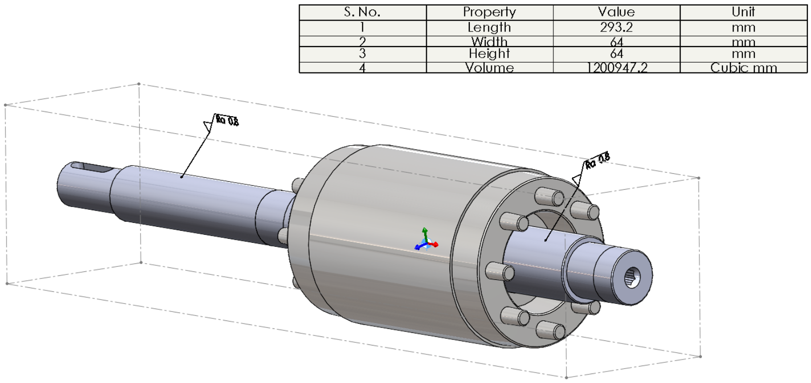
Figure 7. Specification of material density directly as a property of the part.

7.4. Volume and Mass

The measure, product property definition and product property representation schemas from the Integrated Generic Resource: *Fundamentals of product description and support* (Part: 41) can be used for adding various physical properties to the STEP files. The mass value can be directly added to the part using the standard entities. The entities for adding volume are already defined in the standard. The method described in the CAx *Recommended Practices for User Defined Attributes* [49] is used to define the units for volume.

7.5. Bounding Box

There are no entities defined in STEP AP242 standard to add bounding box dimensions directly to the part. A method is proposed in the CAx *Recommended Practices for Geometric and Assembly Validation Properties* [50] for adding a bounding box to the STEP file. The use of two 'CARTESIAN_POINT' entities was recommended to represent the two corner points of the diagonal of the bounding box. We used the same method to add the bounding box of the rotor as shown in Figure 8.



S. No.	Property	Value	Unit
1	Length	293.2	mm
2	Width	64	mm
3	Height	64	mm
4	Volume	1200947.2	Cubic mm

Figure 8. Rotor sub-assembly with the bounding box, volume, surface texture, part coordinate system and center of mass.

8. Constraint Identification

As mentioned in Section 2.2, the object properties are one of the key parameters in grasp planning and gripper selection. In the case of industrial applications, the geometry of the objects is precisely defined during the design phase of the product life cycle. This section proposes a method of reusing the part geometry along with the PMI to identify the object parameters that can be used to specify the constraints for grasp planning and gripper selection.

8.1. Identification of Handling Features from STEP Files

Van Holland et al. [51,52] classified the features as 'handling features' and 'connecting features' and used these features of grasp planning in an assembly process. They further classified the handling features as feeding, fixturing and grasping features. The feeding and fixturing features are used while feeding and initial presentation of components to the feeding area and these features are not available for grasping. This type of classification is not beneficial when the initial feeding and fixturing positions are not defined in the case of bin-picking or where visual sensing is used.

This paper proposes the classification of handling features into the following three categories.

- Forbidden Features: The features that are strictly not available for grasping are referred to as forbidden features. These include the assembly/mating features and the functional features that might be damaged by the grippers, like threaded regions [21], surfaces with special coatings and surface finish requirements. Only special surface finish specifications on particular features are used to identify forbidden features, not the default surface finish requirement that applies to the entire part.
- Restricted Features: The features that can be used during the initial stages of handling tasks like re-orienting and moving from the initial position to the start of assembly engagement are considered as restricted features. For example, in the case of a shaft and housing assemblies both the ends of the shaft are available for grasping in the initial stages of handling. As the shaft starts engaging with the housing during the assembly, one end becomes unavailable for grasping depending on the direction of assembly. Here, such an end of the shaft is a restricted feature and can be used during the initial stages of handling and assembly process. Bounding box dimensions are used to derive the restricted features.
- Grasp Features: The remaining features on the part are open for grasping during all the stages of handling and assembly. These features are called grasp features.

One major criterion in applying the constraints of various part features and gripper selection is that the part should not be damaged during the handling process [12]. The following steps are followed in identifying various handling features from the STEP files.

1. Identify the global coordinate system of the assembly
2. Identify the forbidden features: Features with
   - Assembly constraints
   - Surface texture annotations
   - Thread annotations
   - Coating information
3. Identify the bounding box dimensions of part and mating sub-assembly
4. Convert the bounding box of the part to the global coordinate system
5. Using the mating features, find the final position of the part and its bounding box
6. Find the restricted features, by calculating the interference of the part bounding box with the bounding box of the mating part/sub-assembly
7. The remaining features are identified as grasp features
8. Extract the shape and dimensions of the grasp features
9. Extract/calculate the part properties like density, strength, volume and weight
10. Use these properties for gripper selection, grasp planning and motion planning with collision avoidance

Table 5 shows various types of information from STEP files and which constraint they define.

Table 5. Product information from STEP files and its relevance in grasp planning.

Product Information	Grasp Planning Constraints
Assembly constraints	Forbidden features
Special surface texture annotations	Forbidden features
Thread annotation	Forbidden features
Special coatings	Forbidden features
Bounding box dimensions	Restricted features
Material and material properties	Gripper selection
Volume	Gripper selection
Mass	Gripper selection
Feature dimensions	Gripper selection
Feature shape	Grasp points and gripper selection
Center of Gravity	Selection of grasp points

8.2. Gripper Selection

The object parameters extracted from the STEP AP242 files become input to the selection process of an appropriate gripper for assembling the part. The dimensions of grasp features, part weight and material properties are used to find a gripper. This paper considers only force closure grasp with impactive grippers. A two-level criterion is used in selecting the grippers. The first level narrows down the list of appropriate grippers based on their grasp width and payload. The second level identifies the suitable grippers depending on the maximum gripping force and frictional forces between the gripper and the part. The criterion for selecting the right gripper is as follows.

Level 1:

1. The size of the grasp features (width/diameter) should be between the minimum and maximum grasp width of the gripper.
2. When the minimum grasp width of the gripper is not available, the stroke should be greater than the grasp features' size.
3. The gripper's payload should be greater than the part weight. This paper suggests a maximum limit to the gripper's payload to twice the weight of the part.

Level 2:

1. The gripping force should be able to lift the part only through friction, i.e., the frictional force (F_f) should be greater than the gravitational force [1], as shown in Figure 9.

$$F_g > (m \cdot g)/(\mu \cdot n) \tag{1}$$

where F_g: Gripping force;

$$F_g = \mu \cdot n \cdot F_f \tag{2}$$

m: mass of the object;
g: acceleration due to gravity;
μ: coefficient of friction between the gripper fingers and the object;
n: number of gripper finger.

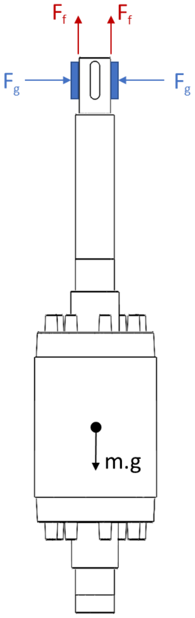


Figure 9. Forces acting on the part during gripping.

Here, the part is considered rigid under the gripping forces. While identifying the grasp regions of the object, the damage to surface finish, threads and coating is already considered and such regions are classified as forbidden regions. Another aspect that can be considered in the selection of the gripper is its weight. If the weight of the gripper is large, it will take a significant portion of the robot's payload. As we have limited the payload of the gripper to a maximum of twice the object weight, this eliminates the possibility of selecting a big/heavy gripper that can impact the robot's payload.

8.3. Grasp Planning/Execution

Once the proper gripper is in place and all the object constraints are identified, the next task will be to plan or execute the grasp. For simple geometries where the gripper is close to the desired grasp location, constraint-based robot programming can be used to create a controller with which the gripper position converges to the grasp position. eTaSL/eTC is a constraint-based task specification system based on Lua for robot programming [53]. The gripping constraints can be easily included in the constraint set defined by Mohammed et al. [39] for completing the assembly operation. For simple geometries where the initial conditions place the robot manipulator further from the grasping location, Somani presents a sampling-based motion planner that samples in the nullspace of geometric constraints [24]. For more complicated geometries where the approach trajectory to the grasp location may require multiple reorientations of the gripper to avoid collision with the environment or forbidden surfaces, a more general motion planning solution such as the solvers from OMPL [54] is required. Evaluation of which solution to use is system architecture and workspace layout dependent and not part of the current work.

8.4. Motor Use Case

This process is applied to a motor assembly. Figure 8 shows the rotor with the bounding box, volume and surface texture annotations along the part coordinate system and center of mass (coinciding with the origin of the part coordinate system). The PMI in this paper does not correspond to the actual design of the part, but hypothetical PMI was added to demonstrate the method. The PMI is added to the housing and rotor parts using JSDAI API [55]. Figure 10 shows the rotor assembled with the motor housing.

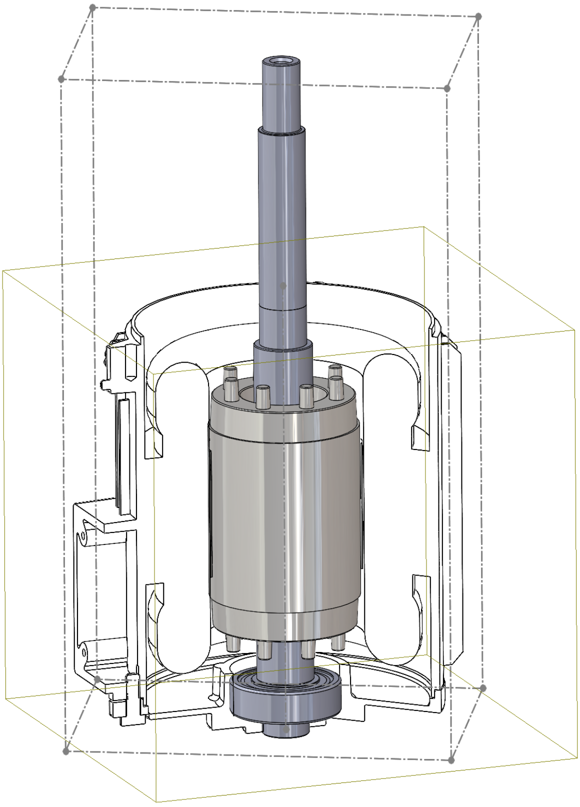


Figure 10. Rotor sub-assembly assembled with Housing showing bounding boxes.

The relevant information is identified and extracted from the STEP files and used in identifying the various types of features on the rotor part.

Figure 11 shows the various types of handling features of the rotor sub-assembly. Only the grasp feature (shown in green) is grasped from starting of handling until the completion of the assembly task. The restricted features can be used for extra support during movement and collision avoidance of the part from the initial position. The forbidden features are identified from the assembly constraints and surface texture annotations. All other PMI such as material is also extracted and used to select the appropriate gripper for the assembly operation.

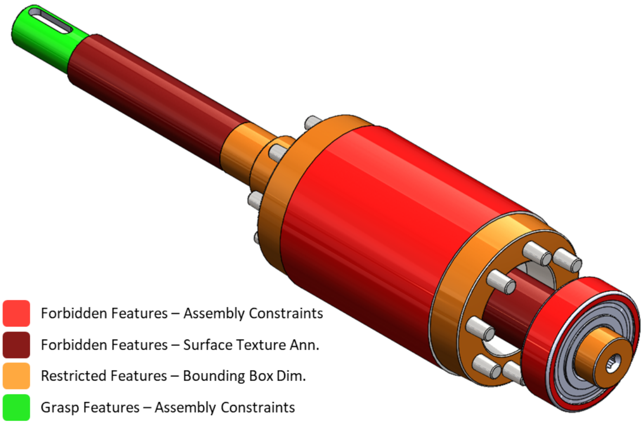

🟥 Forbidden Features – Assembly Constraints
🟫 Forbidden Features – Surface Texture Ann.
🟧 Restricted Features – Bounding Box Dim.
🟩 Grasp Features – Assembly Constraints

Figure 11. Handling features of rotor sub-assembly.

The diameter of the grasp feature is 15.97 mm and the weight of the rotor sub-assembly is 2.88 kg. With these values, the criterion described in Section 8.2 is followed to select a suitable gripper for this assembly operation. A total of 24 two-fingered impactive grippers from four different manufacturers are considered. The summary of these grippers is given in Table 6.

Taking a coefficient of friction of 0.25 for steel–steel contact between the gripper fingers and the part, the minimum gripping force needed is 56.51 N. With these values of weight, diameter and gripping forces, five grippers that match the criterion were selected. This demonstrates that suitable grippers can be selected by using the object parameters and constraints extracted from the STEP AP242 files.

Table 6. Summary of Grippers Considered for the Use Case.

Manufacturer	Number of Grippers	Payload Range (Kg)	Maximum Grasp Width	Stroke (mm)	Max. Gripping Force (N)
OnRobot	5	2–20	430	38–160	450
Shunk	11	0.55–42	620	5–90	8460
Robotiq	3	2.5–5	140	50–140	235
Festo	5	0.2–4.24	40	20–40	3716

9. Conclusions

This paper demonstrated how the relevant product geometric and manufacturing information from STEP AP242 files could be reused in grasp planning and gripper selection. Two ways of adding the surface texture annotation to the STEP files are described. Using these methods, the surface finish annotations are semantically added to the STEP files as per the latest ISO standards on surface texture annotations. Though the STEP AP242 standard is defined to include surface texture information as per ISO 1302:2002, the methods presented in this paper enable the addition of surface texture annotations as per the latest ISO 21920-1:2021 standard. These methods can also be easily adapted to add areal surface texture symbols as per the ISO 25178-1:2016 standard. Other relevant information is added to the STEP files as per the latest STEP AP242 standard that match the recommendations of the CAx implementer forum.

This paper proposes a novel classification of handling features and a method to identify them using the product information from the STEP files. This classification is appropriate for handling and assembly operations and can also be combined with other data-driven or sensor-dependent grasp planning strategies. The rules are used to identify various types of handling features on a rotor of a motor assembly, as demonstrated in the use case. A criterion for selecting the grippers using the object parameters extracted from the STEP AP242 files is presented. This is also demonstrated by selecting appropriate grippers for handling the rotor sub-assembly.

The surface texture annotations are only used in identifying the constrained regions of forbidden features in this paper. However, these annotations are fully semantic and can be used for other manufacturing and downstream processes of surface roughness inspection and quality assurance. The restricted features can be used in supporting and handling large objects using multi-robotic systems during the initial phases or until when those features actually start engaging with the other parts.

Author Contributions: Conceptualization, S.K.M, M.H.A. and L.T.; methodology, S.K.M.; writing—original draft preparation, S.K.M.; writing—review and editing, S.K.M, L.T. and M.H.A.; supervision, L.T.; All authors have read and agreed to the published version of the manuscript.

Funding: The work reported in this paper is supported by the Norwegian Research Council infrastructure project MANULAB: Norwegian Manufacturing Research Laboratory under grant 269898. And partially based on activities within the center for research-based innovation SFI Manufacturing and partially funded by the Research Council of Norway under contract number 237900.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Mjøs Metallvare for providing the CAD model for the use-case example.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ASME	American Society of Mechanical Engineers
CAD	Computer Aided Design
CAE	Computer Aided Engineering
DT	Digital Thread
GD&T	Geometric Dimensioning and Tolerancing
ISO	International Standards Organisation
MBD	Model Based Definition
MBE	Model Based Enterprise
NC	Numerical Controlled
OEM	Original Equipment Manufacturer
PMI	Product Manufacturing Information
SME	Small and Medium Enterprise
STEP	STandard for the Exchange of Product Model Data

References

1. Monkman, G.J.; Hesse, S.; Steinmann, R.; Schunk, H. *Robot Grippers*; WILEY-VCH Verlag GmbH & Co. KGaA: Weinheim, Germany, 2007.
2. Wolf, A.; Steinmann, R.; Schunk, H. *Grippers in Motion: The Fascination of Automated Handling Tasks*; Springer: Berlin/Heidelberg, Germany, 2005.
3. Lundstrom, G. Industrial Robot Grippers. *Ind. Robot.* **1973**, *1*, 72–82. [CrossRef]
4. Dzedzickis, A.; Subaciute-Zemaitiene, J.; Sutinys, E.; Samukaite-Bubniene, U.; Bucinskas, V. Advanced Applications of Industrial Robotics: New Trends and Possibilities. *Appl. Sci.* **2022**, *12*, 135. [CrossRef]
5. Lotter, B. *Manufacturing Assembly Handbook*; VDI-Verlag GmbH: Düsseldorf, Germany, 1986.
6. Boothroyd, G.; Dewhurst, P.; Knight, W.A. *Product Design for Manufacture and Assembly*; CRC Press: Boca Raton, FL, USA, 2011.
7. Mohammed, S.K.; Arbo, M.H.; Tingelstad, L. Constraint Identification from STEP AP242 files for Automated Robotic Welding. In Proceedings of the 2021 IEEE 12th International Conference on Mechanical and Intelligent Manufacturing Technologies (ICMIMT), Cape Town, South Africa, 13–15 May 2021; pp. 277–282.

8. Bohg, J.; Morales, A.; Asfour, T.; Kragic, D. Data-Driven Grasp Synthesis—A Survey. *IEEE Trans. Robot.* **2014**, *30*, 289–309. [CrossRef]
9. Honarpardaz, M.; Tarkian, M.; Olvander, J.; Feng, X. Finger design automation for industrial robot grippers: A review. *Robot. Auton. Syst.* **2017**, *87*, 104–119. [CrossRef]
10. De Souza, J.P.C.; Rocha, L.F.; Oliveira, P.M.; Moreira, A.P.; Boaventura-Cunha, J. Robotic Grasping: From Wrench Space Heuristics to Deep Learning Policies. *Robot. Comput.-Integr. Manuf.* **2021**, *71*, 102176. [CrossRef]
11. Kleeberger, K.; Bormann, R.; Kraus, W.; Huber, M.F. A Survey on Learning-Based Robotic Grasping. *Curr. Robot. Rep.* **2020**, *1*, 239–249. [CrossRef]
12. Owen, A.E. *Flexible Assembly Systems*; Springer Science + Business Media: New York, NY, USA, 1984.
13. Cuadrado, J.; Naya, M.A.; Ceccarelli, M.; Carbone, G. An optimum design procedure for two-finger grippers: A case of study. *IFToMM Electron. J. Comput. Kinemat.* **2002**, *15403*, 2002.
14. Fantoni, G.; Capiferri, S.; Tilli, J. Method for supporting the selection of robot grippers. *Procedia CIRP* **2014**, *21*, 330–335. [CrossRef]
15. Birglen, L.; Schlicht, T. A statistical review of industrial robotic grippers. *Robot. Comput.-Integr. Manuf.* **2018**, *49*, 88–97. [CrossRef]
16. Nguyen, V.D. Constructing Force-Closure Grasps. In Proceedings of the 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 7–10 April 1986; pp. 1368–1373.
17. Reuleaux, F. *Kinematics of Machinery*; (First Published in Germany, 1875); Macmillan and Co.: London, UK, 1876.
18. Bicchi, A. On the Closure Properties of Robotic Grasping. *Int. J. Robot. Res.* **1995**, *14*, 319–334. [CrossRef]
19. Kleeberger, K.; Schnitzler, J.; Khalid, M.U.; Bormann, R.; Kraus, W.; Huber, M.F. Precise Object Placement with Pose Distance Estimations for Different Objects and Grippers. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September 2021–1 October 2021; pp. 4639–4646.
20. Shimoga, K.B. Robot Grasp Synthesis Algorithms: A Survey. *Int. J. Robot. Res.* **1996**, *15*, 230–266. [CrossRef]
21. Van Bruggen, M.; Baartman, J.P.; Bronsvoort, W.F. Grips on Parts. In Proceedings of the Proceedings IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 2–6 May 1993; Volume 2, pp. 828–833.
22. Miller, A.T.; Knoop, S.; Christensen, H.I.; Allen, P.K. Automatic Grasp Planning Using Shape Primitives. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; Volume 2, pp. 1824–1829.
23. Schmalz, J.; Giering, L.; Holzle, M.; Huber, N.; Reinhart, G. Method for the Automated Dimensioning of Gripper Systems. *Procedia CIRP* **2016**, *44*, 239–244. [CrossRef]
24. Somani, N. Constraint-Based Approaches for Robotic Systems: From Computer Vision to Real-Time Robot Control. Ph.D. Thesis, Technical University, Munich, Germany, 2018.
25. Lanni, C.; Ceccarelli, M. An Optimization Problem Algorithm for Kinematic Design of Mechanisms for Two-Finger Grippers. *Open Mech. Eng. J.* **2009**, *3*, 49–62. [CrossRef]
26. Department of Defense, United States of America (USA). *MIL-STD-31000A: Technical Data Packages*; Federal Standardization Manual, Department of Defense: Picatinny Arsenal, NJ, USA, 2009.
27. Herron, J.B. *Re–Use Your CAD: The Model–Based CAD Handbook*; Action Engineering LLC: Golden, CO, USA, 2013.
28. Urbas, U.; Vrabic, R.; Vukasinovic, N. Displaying Product Manufacturing Information in Augmented Reality for Inspection. *Procedia CIRP* **2019**, *81*, 832–837. [CrossRef]
29. Goher, K.; Shehab, E.; Al-Ashaab, A. Model-Based Definition and Enterprise: State-of-the-art and future trends. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2020**, *235*, 2288–2299. [CrossRef]
30. *ISO 10303–242:2014*; Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 242: Application Protocol: Managed Model-Based 3D Engineering. International Organization for Standardization: Geneva, Switzerland, 2014.
31. Kramer, T.; Xu, X. STEP in a Nutshell. In *Advanced Design and Manufacturing Based on STEP*; Xu, X., Lee, A.Y.C., Eds.; Springer Series in Advanced Manufacturing: Advanced Design and Manufacturing Based on STEP; Springer: London, UK, 2009; Chapter 1, pp. 1–22.
32. STEP AP242 Project Committee. Development of STEP AP 242 ed2: "Managed Model Based 3D Engineering. *Whitepaper*; STEP AP242 Project Committee: 2014. Available online: http://www.ap242.org/edition-2 (accessed on 27 November 2022).
33. *ISO 10303–242:2020*; Industrial Automation Systems and Integration—Product Data Representation and Exchange—Part 242: Application Protocol: Managed Model-Based 3D Engineering. International Organization for Standardization: Geneva, Switzerland, 2020.
34. Praveen, O.V.S.; Dileep, B.; Gayatri, S.; Lawrence, D.; Manu, R. Automated Tolerance Analysis of Mechanical Assembly Using STEP AP242 Managed Model-Based 3D Engineering. In *Industry 4.0 and Advanced Manufacturing*; Lecture Notes in Mechanical Engineering; Springer: Gateway East, Singapore, 2021; pp. 149–157.
35. Feeney, A.B.; Frechette, S.P.; Srinivasan, V. A Portrait of an ISO STEP Tolerancing Standard as an Enabler of Smart Manufacturing Systems. *ASME J. Comput. Inf. Sci. Eng.* **2015**, *15*, 021001. [CrossRef]
36. Zhong, R.Y.; Xu, X.; Klotz, E.; Newman, S.T. Intelligent Manufacturing in the Context of Industry 4.0: A Review. *Engineering* **2017**, *3*, 616–630. [CrossRef]
37. Thomas, R.G.; Lawrence, D.; Manu, R. STEP AP242 Managed Model–based 3D Engineering: An Application Towards the Automation of Fixture Planning. *Int. J. Autom. Comput.* **2021**, *18*, 731–746. [CrossRef]

38. *ASME Y14.41:2019*; Digital Product Definition Data Practices. The American Society of Mechanical Engineers: New York, NY, USA, 2019.
39. Mohammed, S.K.; Arbo, M.H.; Tingelstad, L. Leveraging Model Based Definition and STEP AP242 in Task Specification for Robotic Assembly. *Procedia CIRP* **2021**, *97*, 92–97. [CrossRef]
40. Mohammed, S.K.; Arbo, M.H.; Tingelstad, L. Using Semantic Geometric Dimensioning and Tolerancing (GD&T) Information from STEP AP242 Neutral Exchange Files for Robotic Applications. *Int. J. Interact. Des. Manuf. (IJIDeM)* **2022**. *Submitted for publication*.
41. *ISO 21920-1:2021*; Geometrical Product Specifications (GPS)—Surface Texture: Profile—Part 1: Indication of Surface Texture. International Organization for Standardization: Geneva, Switzerland, 2021.
42. *ISO 1302:2002*; Geometrical Product Specifications (GPS)—Indication of Surface Texture in Technical Product Documentation. International Organization for Standardization: Geneva, Switzerland, 2002.
43. *ISO 25178-1:2016*; Geometrical Product Specifications (GPS)—Surface Texture: Areal—Part 1: Indication of Surface Texture. International Organization for Standardization: Geneva, Switzerland, 2016.
44. *ISO 21920-2:2021*; Geometrical Product Specifications (GPS)—Surface Texture: Profile—Part 2: Terms, Definitions and Surface Texture Parameters. International Organization for Standardization: Geneva, Switzerland, 2021.
45. *ISO 21920-3:2021*; Geometrical Product Specifications (GPS)—Surface Texture: Profile—Part 3: Specification Operators. International Organization for Standardization: Geneva, Switzerland, 2021.
46. *ISO 25178-2:2021*; Geometrical Product Specifications (GPS)—Surface Texture: Areal —Part 2: Terms, Definitions and Surface Texture Parameters. International Organization for Standardization: Geneva, Switzerland, 2021.
47. *ISO 25178-3:2012*; Geometrical Product Specifications (GPS)—Surface Texture: Areal—Part 3: Specification Operators. International Organization for Standardization: Geneva, Switzerland, 2012.
48. CAx Implementor Forum. PMI Unicode String Specification Examples and Mapping Strategies—for Dimensioning and Tolerancing, GD&T, Surface Texture Symbol and Welding Symbol PMI Annotation Entities. *Revision J*. 2011. Available online: https://www.mbx-if.org/documents/rec_prac_unicode_strings_rev_j_2011-05-25.pdf (accessed on 27 November 2022).
49. CAx Implementor Forum. Recommended Practices for User Defined Attributes. *Release 1.3*. 2014. Available online: https://www.mbx-if.org/documents/rec_prac_user_def_attributes_v13.pdf (accessed on 27 November 2022).
50. CAx Implementor Forum. Recommended Practices for Geometric and Assembly Validation Properties. *Release 4.4*. 2016. Available online: https://www.mbx-if.org/documents/rec_prac_gvp_v44.pdf (accessed on 27 November 2022).
51. Van Holland, W.; Bronsvoort, W.F. Assembly Features in Modeling and Planning. *Robot. Comput. Integr. Manuf.* **2000**, *16*, 277–294. [CrossRef]
52. Van Holland, W. Assembly Features in Modeling and Planning. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 1997.
53. Schutter, J.D.; Laet, T.D.; Rutgeerts, J.; Decre, W.; Smits, R.; Aertbelien, E.; Claes, K.; Bruyninckx, H. Constraint-based task specification and estimation for sensor-based robot systems in the presence of geometric uncertainty. *Int. J. Robot. Res.* **2007**, *26*, 433–455. [CrossRef]
54. Şucan, I.A.; Moll, M.; Kavraki, L.E. The Open Motion Planning Library. *IEEE Robot. Autom. Mag.* **2012**, *19*, 72–82. [CrossRef]
55. LKSoftWare GmbH, Germany. JSDAI. Available online: https://www.jsdai.net/ (accessed on 27 November 2022).

A.5. Paper V:

Shafi K. Mohammed, Lars Tingelstad. (2023) “Robotic Assembly Using Product Manufacturing Information from Model Based Definition,” (Will be submitted)

Appendix B.

Introduction to STEP Standard

This Appendix presents the overall structure of the STEP standard along with a brief description of its critical methods that enable the description of various elements, the definitions guiding the organization of data inside a STEP file, implementation, and conformance classes.

B.1. Structure of STEP Standard

An Application Protocol (AP) is the highest level of information model that can be implemented for a particular industrial application. From its beginning, the structure of the STEP standard evolved from a monolithic model to an extended

B.1.1. Monolithic Architecture

Initially, STEP started with a 'monolithic' (non-modular) architecture where each AP specifies a complete data definition on its own. The first AP developed (AP203) is a non-modular AP. As shown in Fig. B.1, each AP consists of an Application Activity Model (AAM), Application Reference Model (ARM), Application Interpreted Model (AIM), and Implementation Method (IM). Application Interpreted Constructs (AICs) facilitate data exchange between different APs using AIM [6].

B.1.2. Modular Architecture

As the number of APs in STEP increased, the need and opportunity for reusing the data models in multiple APs increased. To enable interoperability of data

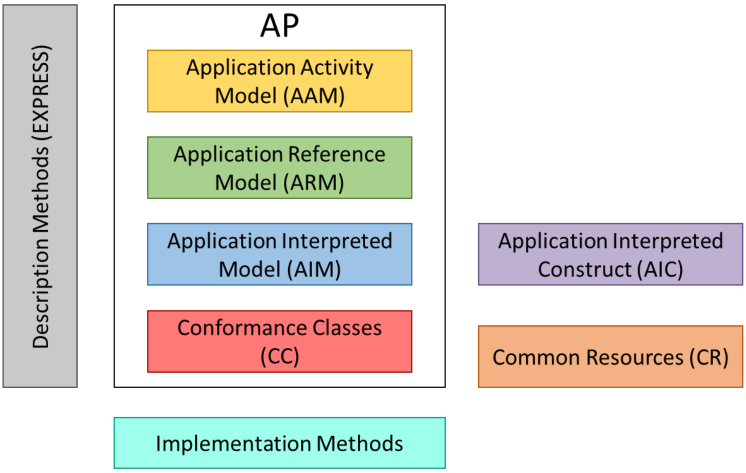

Description Methods (EXPRESS)
AP
Application Activity Model (AAM)
Application Reference Model (ARM)
Application Interpreted Model (AIM)
Conformance Classes (CC)
Application Interpreted Construct (AIC)
Common Resources (CR)
Implementation Methods

Figure B.1.: Components of in the initial non-modular architecture of STEP

models among APs, modular architecture is adopted by STEP. Fig. B.2 shows the structure of a modular STEP standard. The data model is divided into smaller reusable modules called 'Application Modules (AM)'. Each AP has its own AM known as 'Protocol AM' which has the data specification for that particular AP. There are other AMs that can be shared/referred by multiple APs. Each AM has its own ARM and AIM. AP203Ed2 is the first modular AP in the STEP standard.

B.1.3. Extended Architecture

The modular architecture did not achieve full AP interoperability. Another aspect that can improve the adoption of the standard is domain interoperability and making the standard accessible to domain expert [8]. To achieve this objective, AP242Ed1 and AP209Ed2 are extended by adding the Business Object Model (BO Model). An extended architecture is developed based on Core Model (CM) and Domain Model (DM). CM divides the product data model into Core Technical Capabilities (CTC) that can be shared by all the AP Domain Models (DM). The CTCs can be specialized for the DM of any AP. Some of the CTCs identified specifically for AP242 are 3D Geometry, Shape Association and Structure, Kinematics, Mating, and Material. The CM is not included in the second edition of AP242 (AP242Ed2). A DM is specified in AP242Ed2, which maps directly to the Application Reference Model (ARM) as the CM is unavailable. The third edition of AP242 (AP242Ed3) is planned to have a fully extended architecture by integrating CM [58]. The architecture of AP242Ed2 is partially extended with a DM [48, 113]. The extended architecture of AP242Ed2 is shown in Fig. B.3.

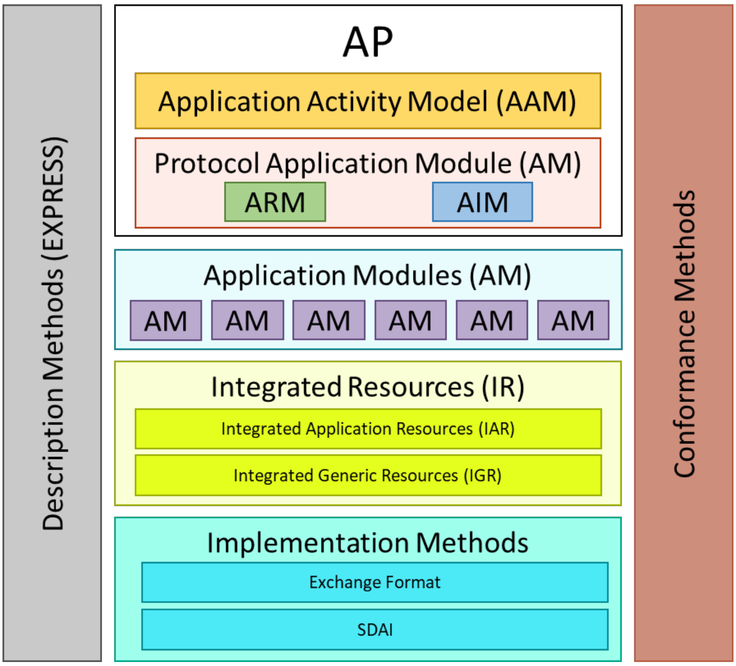

Description Methods (EXPRESS)
AP
Application Activity Model (AAM)
Protocol Application Module (AM)
ARM
AIM
Application Modules (AM)
AM
AM
AM
AM
AM
AM
Integrated Resources (IR)
Integrated Application Resources (IAR)
Integrated Generic Resources (IGR)
Implementation Methods
Exchange Format
SDAI
Conformance Methods

Figure B.2.: Modular architecture of STEP

This architecture improves the interoperability among various APs throughout the product life cycle support. The major elements of STEP architecture are

- Description Methods
- Business Object Model (BO Model)
- Domain Model (DM)
- Application Module (AM)
- Integrated Resources (IR)
- Implementation Methods
- Conformance Methods

B.1.4. Description Methods

The data models of STEP are specified using description methods. A formal information modeling language known as 'EXPRESS' is developed for this specific purpose [26, 42]. EXPRESS is defined in Part-11 of ISO 10303. It is both human-understandable and computer-interpretable language. It is an 'entity-oriented'

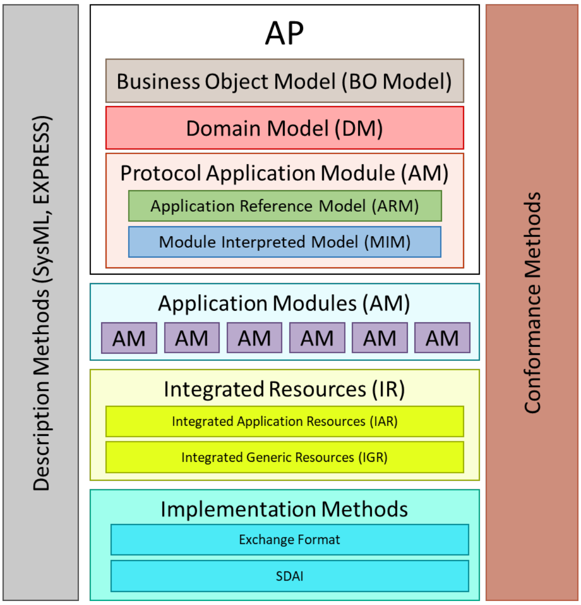

Description Methods (SysML, EXPRESS)
AP
Business Object Model (BO Model)
Domain Model (DM)
Protocol Application Module (AM)
Application Reference Model (ARM)
Module Interpreted Model (MIM)
Application Modules (AM)
AM
AM
AM
AM
AM
AM
Integrated Resources (IR)
Integrated Application Resources (IAR)
Integrated Generic Resources (IGR)
Implementation Methods
Exchange Format
SDAI
Conformance Methods

Figure B.3.: Extended architecture of STEP.

language (comparable to object-oriented) independent of programming languages and database systems [105]. EXPRESS-G is a graphical description method based on EXPRESS used by STEP.

With the extended architecture, the information specifications are given in SysML, a modeling language used in Model Based Systems Engineering (MBSE) [121]. This is a move towards connecting Systems Engineering (SE) with STEP and bringing SE interoperability.

Listing B.1: EXPRESS Entity Definition from IGR ‘Geometric and topological representation’ (Part-42) [48].

```

ENTITY cartesian_point
  SUPERTYPE OF ( Omitted for simplicity )
  SUBTYPE OF ( point );
  coordinates : LIST[1:3] OF length_measure;
END_ENTITY;

ENTITY direction
  SUBTYPE OF ( geometric_representation_item );
  direction_ratios : LIST[2:3] OF REAL;
WHERE
  WR1: Omitted for simplicity ;
END_ENTITY;

```

```
ENTITY placement
  SUPERTYPE OF ( Omitted for simplicity )
  SUBTYPE OF ( geometric_representation_item );
  location : cartesian_point;
END_ENTITY;

ENTITY axis2_placement_3d
  SUBTYPE OF ( placement );
  axis : OPTIONAL direction;
  ref_direction : OPTIONAL direction;
DERIVE
  Omitted for simplicity;
WHERE
  WR1: Omitted for simplicity ;
  WR2: Omitted for simplicity ;
  WR3: Omitted for simplicity ;
  WR4: Omitted for simplicity ;
END_ENTITY;

ENTITY volume
  SUPERTYPE OF ( Omitted for simplicity )
  SUBTYPE OF ( geometric_representation_item );
WHERE
  WR1: Omitted for simplicity ;
END_ENTITY;

ENTITY cylindrical_volume
  SUBTYPE OF (volume);
  position : axis2_placement_3d;
  radius : positive_length_measure;
  height : positive_length_measure;
END_ENTITY;
```

Listing B.1 shows the entity definitions of some of the geometric elements for specifying a cylindrical object. Fig. B.4 shows the SysML diagrams for the same entities defined in the listing B.1.

B.1.5. BO Model

BO Model is defined to communicate the major concepts and information requirements of the AP with the domain experts. BO Model is defined at a higher granularity than the Application Reference Model (ARM) of the APs and offers better communication with domain experts [8]. The BO Model is specified using SysML and maps to the Application Reference Model (ARM) of the AP. The implementation and data exchange can be achieved by deriving an XML schema from the BO Model.

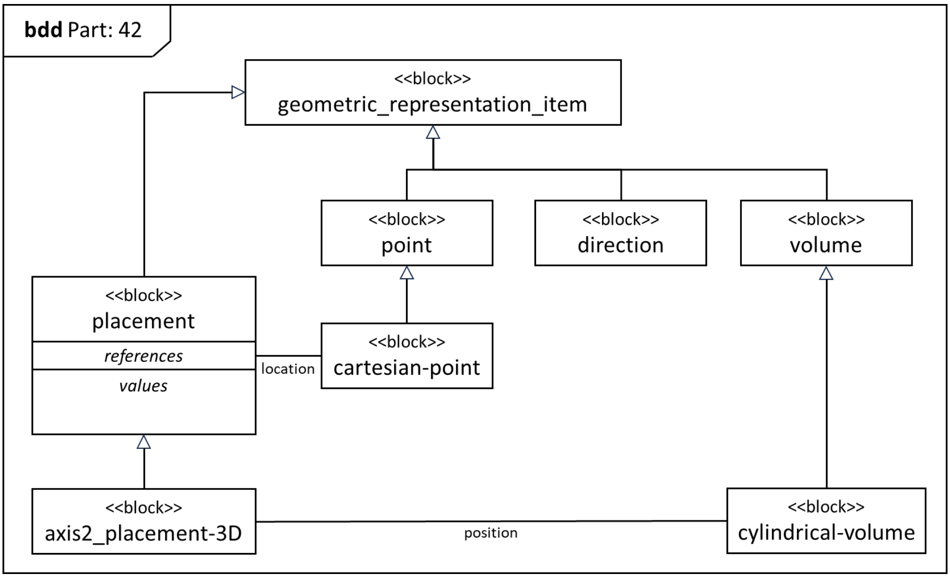


Figure B.4.: SysML Block Definition Diagram showing some of the entities of 'Geometric and topological representation' (Part-42).

B.1.6. Domain Model

AP242Ed2 has a Domain Model (DM) along with the BO Model. The DM specifies the information requirements in terms of Domain Objects (DO) and maps them to the ARM. The DM presents the information at medium-granularity to communicate with the domain experts. It is defined using SysML, and XML schemas can be derived from the BO Model and the DM.

B.1.7. Application Module (AM)

AMs are 'bite-sized' information specifications that can be implemented and referred by multiple APs. Each AM has an Application Reference Model (ARM) and a Module Interpreted Model (MIM).

- ARM: This presents the information requirements in the domain terminology for an application context. The entities defining these requirements are known as 'application elements.'
- MIM: This specifies the implementation model of the AM. The requirements given in ARM are interpreted and mapped to Integrated Resources. The integrated resources can be extended to match the information requirements of the AM. These extended mappings are available for all other AMs for reuse.

Each AP has a specific AM, known as the ‘AP Module,’ that captures its information requirements and presents the corresponding MIM. The AP Module has the same name as the AP, and its part number is arrived at by adding 200 to the AP part number (i.e., Part number of AP Module = 200 + Part number of AP). The MIM from the AP Module and the XML schema from the DM are the implementation models.

B.1.8. Integrated Resources

The integrated resources define the basic elements for capturing the product data. These specifications are independent of application and domain, and form the building blocks for the APs. Depending on the application and domain, the integrated resources are interpreted and extended for implementation in the MIMs [72, 105]. There are two types of integrated resources,

- Integrated Application Resources (IAR): These resources specify the entities used mainly by the AP.
- Integrated Generic Resources (IGR): These types of resources specify the entities that could be used in many APs. These are at the bottom layer of the data specification hierarchy of an AP and could not reference the entities defined in the higher level [80].

B.1.9. Implementation Methods

Implementation methods specify the techniques and structures of product data storage and exchange. The implementation methods are given in Part: 21 to 28 of the standard. These parts are as follows.

- Part-21: This part specifies the exchange structure using a physical file based on clear text encoding [43]. These are called ‘STEP files’ or ‘Part21 files’ or ‘physical files’ and have an extension of ‘.stp’ or ‘.step’. Listing B.2 shows a sample STEP file showing the geometric entities for specifying a rectangular prism.
- Part-22: This document ‘Standard data access interface’ (SDAI) specifies a language-independent API to access EXPRESS-based data models and databases. The SDAI API allows the separation of data exchange software from the applications using the STEP data [44, 105].
- Parts-23, 24, and 27: These parts present the C++, C, and Java language bindings of SDAI. JSDAI API is a Java implementation of SDAI using Part-27 [45, 46, 61]. JSDAI Express Compiler is used for this thesis to parse the EXPRESS schemas from AP242Ed2.

- Part-25: This part of the standard specifies a mapping from EXPRESS to Unified Modeling Language (UML) [59] and facilitates the generation of XML Metadata Interchange (XMI) files from EXPRESS data models. XMI is a metadata exchange standard developed by Object Management Group (OMG) [122].
- Part-26: A binary file exchange structure for EXPRESS data using Hierarchical Data Format Version 5 (HDF5) [120] is specified in this part [60, 80].
- Part-28: This part gives the XML representation of EXPRESS data models [49].

Listing B.2: STEP file showing the entities for defining the geometry of a rectangular prism object.

```
#12=CARTESIAN_POINT('',(0.0,0.0,0.0));
#13=DIRECTION('',(0.0,0.0,1.0));
#14=DIRECTION('',(1.0,0.0,0.0));
#15=AXIS2_PLACEMENT_3D('',#12,#13,#14);
#30=AXIS2_PLACEMENT_3D('',#27,#28,#29);
#31=PLANE('',#30);
#32=CARTESIAN_POINT('',(45.0,0.0,30.0));
#33=VERTEX_POINT('',#32);
#36=CARTESIAN_POINT('',(45.0,0.0,30.0));
#37=DIRECTION('',(0.0,0.0,-1.0));
#38=VECTOR('',#37,60.0);
#39=LINE('',#36,#38);
#40=EDGE_CURVE('',#33,#35,#39,.T.);
#41=ORIENTED_EDGE('',*,*,#40,.F.);
#63=ORIENTED_EDGE('',*,*,#62,.F.);
#64=EDGE_LOOP('',(#41,#49,#57,#63));
#65=FACE_BOUND('',#64,.T.);
#66=ADVANCED_FACE('',(#65),#31,.F.);

#175=CLOSED_SHELL('',(#66,#97,#121,#145,#162,#174));
#176=MANIFOLD_SOLID_BREP('Boss-Extrude1',#175);

#16=SHAPE_REPRESENTATION('Default',(#15),#10);
#17=ADVANCED_BREP_SHAPE_REPRESENTATION('Default',(#176),#10);
#185=SHAPE_REPRESENTATION_RELATIONSHIP('','',#16,#17);

#192=APPLICATION_CONTEXT('Managed model based 3d engineering');
#193=APPLICATION_PROTOCOL_DEFINITION('international standard','ap242_managed_model_based_3d_engineering',2013,#192);
#194=PRODUCT_CONTEXT('',#192,'mechanical');
#195=PRODUCT_DEFINITION_CONTEXT('part definition',#192,'design');
#24=PRODUCT('Default','Default','',(#194));
#196=PRODUCT_RELATED_PRODUCT_CATEGORY('part','',(#24));
#197=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE('','',#24,.NOT_KNOWN.);
#25=PRODUCT_DEFINITION('design','',#197,#195);
#23=PRODUCT_DEFINITION_SHAPE('','',#25);
#198=SHAPE_DEFINITION_REPRESENTATION(#23,#16);

#1=MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION('',(#26),#10);
```

B.1.10. Conformance Methods

The conformance methods define the methodology to test the implementations of the APs to establish the conformance with the standard. Conformance methods

Table B.1.: Elements of STEP Standard and their part numbers [41]

Elements of STEP Standard	Part Numbers
Description Methods	11–19
Implementation Methods	21–29
Conformance Testing Methodology and Framework	31–39
Integrated Generic Resources (IGR)	41–99
Integrated Application Resources (IAR)	101–199
Application Protocol (AP)	201–299
Abstract Test Suites	301–399
AP Modules	401–499
Application Modules (AM)	1001–1999
Business Object Models (BO Model)	3001–3099
Domain Model (DM)	4401–4499
Usage Guides	5001–5099

include conformance testing methodology and framework and abstract test suites. ISO 10303-30 series deal with testing methodology and framework, and 300 series parts specify abstract test suites [72, 105]. The part numbers from the 300 series correspond to the AP part numbers (200 series) [41].

- Part-31: *Conformance testing methodology and framework: General concepts* [50]
- Part-32: *Conformance testing methodology and framework: Requirements on testing laboratories and clients* [51]
- Part-34: *Conformance testing methodology and framework: Abstract test methods for application protocol implementations* [52]
- Part-35: *Conformance testing methodology and framework: Abstract test methods for standard data access interface (SDAI) implementations* [66]
- Part-304: *Abstract test suite: Mechanical design using boundary representation* [62]
- Part-307: *Abstract test suite: Sheet metal die planning and design* [63]
- Part-325: *Abstract test suite: Building elements using explicit shape representation* [64]
- Part-332: *Abstract test suite: Technical data packaging core information and exchange* [65]

Table. B.1 gives the elements of ISO 10303 standards and their part numbers [41].

ISBN 978-82-326-7524-1 (printed ver.)
ISBN 978-82-326-7523-4 (electronic ver.)
ISSN 1503-8181 (printed ver.)
ISSN 2703-8084 (online ver.)

NTNU
Norwegian University of
Science and Technology