Communication of Design Data in Manufacturing Democratization

Bhairavsingh Ghorpade (Bhairavsingh.Ghorpade@ou.edu)
The University of Oklahoma - Norman Campus: The University of Oklahoma https://orcid.org/0000-0001-9591-9631
Shivakumar Raman

Research Article

Keywords: Manufacturing democratization, STEP file, CAD design, Part dimensional data extraction

Posted Date: May 11th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-460336/v1

Abstract

Part design is the principal source of communicating design intent to manufacturing and inspection. The design data is often communicated through CAD systems. Modern analytics tools and artificial intelligence integration into manufacturing has significantly advanced machine recognition of design specification and manufacturing constraints. This paper is aimed at the collaboration among multiple vendors across supply chains to enable efficient order procurement. To this end, the paper discusses the development of a simple framework for extracting the dimensional data from part design and storing them for enhancing machine readability of the part design at multiple levels of manufacturing.

1. Introduction

Design data is created and maintained across the industries in various computer-aided design (CAD) systems and formats. Various types of CAD systems are being used for designing the part. Throughout a product's lifecycle, CAD files store various product information such as geometrical dimensions and tolerances [1]. The design data plays an important role during the manufacturing of the product. Design data provides insights and milestones during manufacturing [2].

The modern manufacturing industry relies heavily on concurrent manufacturing [3]. Complex products are often manufactured in pieces, from around the globe. Various levels of manufacturing participants contribute to such products. As shown in figure 1 low-level individual parts are manufactured from sheet metal or bar metal and such individual parts are assembled to form the sub-assemblies and such sub-assemblies are assembled further to manufacture the useful product. As the levels in collaborative manufacturing increase, the complexity of the process tends to increase. Each level has restricted mobility in terms of design, production rate, and cost for manufacturing because the output at each level output could be input to the next level, and changes in any design can hamper further assemblies. Hence, engineering change management plays an important and critical role in cost and time to market. Usually, the lower-level participants are small industries and they are often resource-strapped. The procurement initiated by larger companies and the communication for this procurement is still human-centric. Often, after manual verification of the part design, the awarding of a subcontract, scheduling, and sequencing are performed. This makes procurement highly human intervention dependent, leading to many current challenges in real supply chains.

Due to market competition, organizations are often reluctant to share their valuable data with other industries, suppliers, and sub-suppliers. But a need for a common information pool is equally important for small manufacturing businesses that are the backbone of larger organizations. Small businesses stimulate parallel manufacturing and bring down the time to market along with cost. Small businesses are often the drivers of the manufacturing process. Due to insecurity in data sharing, small-scale manufacturers cannot connect with large organizations, restricting the business expansion of the small industries. On other hand, large-scale manufacturers often experience a shortage of proper capacity for manufacturing specialized or customized components for their assemblies. This could result in larger

lead times and costs, by relying only on known suppliers (only those with prior history). Manufacturing democratization allows for larger manufacturers to communicate with a wider set of suppliers.

This has increased the data-sharing aspects between the different manufacturers. CAD files are the primary source for sharing the part design data between different elements of the manufacturing ecosystem. Sharing the design data entertains many challenges such as the data security and redundancy of design data. Mainly, the abundance of design data stored in a solid format often stimulates data redundancy. These CAD models are stored and accessed as 2D or 3D CAD models. Such CAD models and files often require human interaction for understanding the part features and dimensions. Machine readability of such CAD models is being explored largely for applying cutting-edge predictive modeling and artificial intelligence solutions. Manufacturing democratization necessitates online procurement across longer supply chains. In such cases, the exchange of design data files across companies becomes very important to solicit manufacturing services. This paper proposes a method for extracting the feature information along with the dimensional information of individual data from CAD files and its storage of the interlinked data into a machine-readable format such as XML files.

Literature Survey

The STEP structure representation of parts is used for recognizing milling features such as slots, and corner blind slots from a given part [4]. The algorithm used in prior works utilizes the inter-face representation of the part, and features are identified using heuristics. Furthermore, there are a few approaches to recognition of rotational part features such as cylindrical holes, conical holes, cylindrical and conical external features using STEP file [5][6][7]. In these algorithms, the coordinate points are used to identify these features. All the same, the generation of the part geometry's STEP knowledge base plays could play an important role in understanding the dimensionality of the part. Primarily, the manufacturing machines consider the part's dimensions to a given machine's workspace [8]. This requires the part dimensions of the each-sided feature. The machine workspace should be larger than the given part's largest dimension. Undertaking such crucial information helps in identifying the dimensionality constraints of the parts.

Step Files

The STEP format is a neutral CAD format that was developed to standardize the exchange of geometric data under the ISO 10303 format. A STEP file can also be called a part 21 file as the format comes under ISO 10303-21. The STEP format can be used to share not only geometric data but also product information, which could be used for other components of the system.

The main and effective advantage of STEP files is that the STEP file is machine and human-readable. To store design data in a logical order, STEP files are organized using a specific format. Understanding the structure of the STEP file is important for developing the algorithm for extracting the dimensions and feature information of the part. Moreover, STEP files are organized using various keywords, these

keywords navigate the structure and part information throughout the file body. Broadly, the STEP file is organized in two separate sections[5]. These sections are explained below.

1. ***Header Section:*** The header section is located at the top (starting) of the STEP file. This section includes meta-information of part and design file such as the type of STEP file, filename, software used to create the drawing. The section begins with the keyword **HEADER** and ends with **ENDSEC**.
2. ***Data Section:*** Data section is located under the header section. The Data section starts with the keyword **DATA** and ends with **ENDSEC**. This section stores part dimensional and geometrical information such as face numbers, edge numbers, coordinate points, and vertex points of each edge. The information in the data section is represented in a line-wise manner. Each line encloses certain information often called an entity, and the type of information is identified using keywords present in each line called Entity Type. Each line begins with a line number. The line number starts with 1 and has a prefix as '#'. The representation format of the line in the data section is given as follows.

LINE_NO = ENTITY_TYPE (ENTITY)

The STEP file format contains several predefined entity types, and each entity type has a predefined keyword. These entities and their predefined keywords are explained in the next section. Most types of entities store line numbers of the next consecutive entity type. Some entity types store dimensional information such as coordinate points or geometrical information of edge.

The entities of STEP files hierarchically store the information. Each entity guides to the next lower entity and so on until the node of the hierarchical tree is reached. These nodes store various information, but information of principal interest are coordinate point locations. The part design information is organized arbitrarily throughout the STEP file. This makes it hard and convoluted at first glance. However, the STEP file possesses an underlying structure that leads to the extraction of the geometrical features. This extraction is possible with the help of the tree structure, which is explained in the following section.

STEP file possesses a hierarchical tree structure for storing design data and the data is represented in a line-by-line manner. The structure is organized using keywords. These keywords describe the nature of data present in the line in the STEP file. While describing part design in a STEP file, each of these keywords navigates the information of the next hierarchical keyword using the line numbers. These keywords are defined below [9].

1. ***CLOSED_SHELL:*** Closed shell can be considered as a gateway for part design in a STEP file. Every part is given under one closed shell. STEP file divides the part in a face wise section. Each face gets one branch, and it expands further until the vertex point level of each of its edges. The line containing CLOSED_SHELL as a keyword discloses the line numbers of all the ADVANCE_FACEs which bounds the part together.
2. ***ADVANCED_FACE:*** An advanced face is a generic face of a given part. The number of advanced faces depends on the number of faces that enclose the given part. The line with the keyword ADVANCE_FACE contains further line numbers of FACE_OUTER_BOUND/FACE_BOUND. Each

Advance face contains one face outer bound. Hence, only one line number of respective face outer bound is given in this line. Moreover, this line also gives the surface type information such as “CYLINDER” for circular or curved face, “PLANE” for planar surfaces, “CONICAL” for a conical type of surfaces, and “TOROIDAL” for toroidal surfaces.

3. ***FACE_BOUND/FACE_OUTER_BOUND:*** Face outer bound, or face bound both discloses the information of edges which develops the respective face. Each face is formed with a loop of edges. The line with the keyword FACE_BOUND navigates to the EDGE_LOOP lines for further travel.
4. ***EDGE_LOOP:*** Each edge loop is created using several edges. The line with the keyword EDGE_LOOP provides the line numbers for all the edges which disclose the represented surface. These edges are shown in a line with the keyword ORIENTED_EDGE.
5. ***ORIENTED_EDGE:*** The line with the keyword ORIENTED_EDGE directs to the line which contains EDGE_CURVE.
6. ***EDGE_CURVE:*** The line with the keyword EDGE_CURVE points to the line containing the keywords VERTEX_POINT and the type of edge curve. The type of edge curve provides information about the type of the edge such as “CIRCLE” for curves, or “PLANE” for straight edges. (P Sateesh; 2017). This line can be representing different faces. The faces which share edges can be tracked down using EDGE_CURVEs. All entities from and below EDGE_CURVE in the STEP hierarchy are repeated for multiple faces which share the edges.
7. ***VERTEX_POINT:*** The line with the keyword VERTEX_POINT defines the vertex of the edge. This directs towards a line that contains the keyword CARTESIAN_POINT.
8. ***CARTESIAN_POINT:*** The line with the keyword CARTESIAN_POINT is the lower-level entity in the STEP Hierarchical structure. This is the building block of the entire 3D model. These define the position of each point in the three-dimensional space.

Data Extraction Model

The design data primarily consists of the dimensions of the part features. These dimensions are always coupled with the feature orientation and location concerning the other part features. The proposed algorithm and the storage schema utilize these two aspects of design information for proposing and enhancing a new design data storage system. Dimensions of part features, in other words, are the dimensions of the edges of the part features. This is extensively represented in the borderline representation of the given part design.

In the proposed model for dimensional data extraction, the dimensions of such edges are used to systematically extract the part feature dimensions. The part design files are stored in CAD formats. There are various formats available based on the software used to create the drawings. The proposed model uses a universal native CAD STEP AP203 format file for dimensional data extraction.

The part’s feature is the generic shape of the part. The extraction of actual dimensions of any part can be carried using the hierarchical tree structure of STEP files. The STEP file is compiled with two types of the

part's geometric information: face orientation information and face location coordinate points in the CAD workspace environment. For extracting the dimensions these two types of data need to be systematically processed. These extracted dimensions are specifically related to the faces (features) of the parts. These dimensions can be extracted using a specially developed algorithm. The proposed algorithm as shown in **figure 2**, takes the STEP file as input and outputs the face dimensions including radiuses of circular sections. Each advance face is to be processed to get the Cartesian points of the borders of the face. This involves the stepwise processing of each STEP file entity of the advanced face. The main STEP file entities as discussed in the previous section, are FACE_OUTER_BOND, EDGE_LOOP, ORIENTED_EDGE, and EDGE_CURVE. Each face has multiple edges, and these numbers of edges are equal to the number of Edge Curves in the STEP file for the given face. Each Edge Curve has three attributes, in which two are vertex points of a given edge, and the third attribute discloses the nature of the edge, which is either circular or linear.

The circular attribute discloses the radial dimension of the edge (radius). Since the linear edge often does not contain any important information within it, the linear dimensions of the edge can be extracted using Cartesian points. Each vertex point contains the Cartesian point of the edge. Hence, all vertex points of the edge are used to calculate the distance between them.

For calculating the distance, the three-dimensional Euclidean distance formula is used [10]. If P_1 and P_2 be Cartesian points of two adjacent vertex points (V_1, V_2) of edge curve (E_c), then $d(P_1, P_2)$ is the linear distance between vertex points. This linear distance between co-ordinate points (x1, y1, z1) and (x2, y2, z2) is given as: **see formula 1 in the supplementary files.**

Further, the extracted dimensional data from the CAD file is stored into a feature-based logical structure in the XML file. In the XML file syntax, elements are the basic components. These elements contain data bounded by the markup tags. The data stored in these elements are represented by the markup tags and stored data can be numeric or text [11]. In the XML terminology, the information within each element is called content. Content can have again a sub-element or data stored in it. If the content of an element or sub-element contains again a sub-element, then such data storage can be called nested storage. This nested storage helps store data in a hierarchical structure. **Figure 3** shows a proposed structure describing a part.

The expression <Assembly_Name:> and </ Assembly_Name:> are the markup tags for main element. Each XML file represents one assembly. Assembly element further has sub-elements named as <Component_1>, <Component_2>, <Component_3> and so on. These components are basic components of a given assembly. If the extracted dimensions represent a single component and not an assembly, then the main assembly element contains only one component named <Component_1>. Each component sub-element stores another sub-element named 'Features'. Here, <Feature_1 :> and </Feature_1:> are start and end markup tags, respectively. These are also called markups. Each feature contains a unique and separate tag. The tag number is given based on the unique integer (line number) where the specific feature is represented in the STEP file. Further, feature sub-element stores, the dimensions of the given

feature as sub-elements of features. The start and end tag for such sub-elements can be of two types ***<Dimensions_Linear Length = _____ />*** and ***<Dimensions_Circular Radius = _____ />***. The former sub-element represents linear dimensions. Whereas later represents radial dimensions. Each dimensional sub-element stores the data, this data can be numerical or text. In the proposed structure, numeric dimensional values are stored in the sub-elements. For example: <Dimensions_Linear Length = 20 />.

An XML file is a document in which entire assembly data is being stored. In the proposed database format, each assembly is considered as an object which is identified by the 'assembly key'. This key encompasses entire assembly information. Under such assembly, various components can exist. Each component has a unique component key. This component key further has various features. Each feature is recognized by a feature key and the feature key contains specific dimensioning information of the component. The features are automatically recognized by the above algorithm explained and based on the feature the XML file creates a new element for respective features.

Example Implementation

For the demonstration, an example part of the connector is shown in **figure 4**. The part material and other specific information are not considered for the demonstration. The design and dimensions of part features are the important segments for the study.

A Python programming language-based application is developed for dimension extraction. Using the algorithm all dimensions of the part are extracted. A screenshot of the output is shown in figure 5. The face shown in output is a semi-cylindrical internal face of through-hole with a radius of 2.40 inches and a length of 3.00 inches. The face is highlighted in figure 6. Face outer bond 315 in figure 5, represents the unique integer of the given face in the STEP file. The proposed algorithm uses the same unique integer for representing the face in the extracted output. This numbering of the face helps in tracking the dimensions down and linking the faces with adjacent faces for defining complete features.

The extracted dimensions are further stored into an XML file in feature-based logic. Feature-based logical data structure for storing dimensional data fulfills the application requirements. Since most face edges are shared with adjacent faces, face level hierarchy helps in reducing the redundancy of dimensional data.

For demonstrating the data structure and the storage schema, data extracted from Step 4 is used for creating an XML file. **Figure 7** shows the snap of an XML file. The complete file is shown in the appendix.

As shown in **figure 4**, the connector part is a single component and not an assembly. Hence, the XML file only has component_1. Within component_1, various features are given. Each feature is represented by the faces that the feature is made of. For instance, the internal face shown in **figure 6** is represented in Feature_315. The hole (internal face) has a radius of 2.4 inches and a depth of 3 inches. Every feature is represented using a similar representation form. This XML file is generated using an extension for the

algorithm. In this connector example, python language is used for generating the XML file after extracting the feature-based dimensions from the STEP file.

Discussion On Application

Demands for parts are usually generated based on shortages accepted by managers, such that sub-contracts can be awarded for their manufacture. Contracts are based on making designs available and establishing supply chain networks for timely, economic delivery. In manufacturing democratization, even small manufacturers can set demands for order delivery by other small manufacturers. It is often a process of search and discovery [12]. The technical details can be provided using a resource as suggested in this paper, whereby a product to process mapping can assist in the procurement of specific manufacturing services. Thus, this paper provides the interface between demand and supply for manufacturing democratization.

Part features are important elements of manufacturing parts. Often the geometric shape, dimensions, the orientation of such part features play a significant role in defining the manufacturing processes and in planning the processing sequence. Design specifications and product manufacturing information of part features are usually implicitly and explicitly embedded in product models and design drawings, and these are important to extract while selecting processes and machines. In automated process planning systems, different algorithms are often used for understanding the part feature information [13][14]. This paper provides the interface between design features and product manufacturing through simple XML files. Machine Learning and AI can thus be utilized to detect product similarities and process sequencing.

As shown in figure 8, let us consider a part with three different types of manufacturing features: external cylindrical feature, rectangular feature, and a hole feature. Each of these three features requires a different set of manufacturing processes. These can be extracted using the procedure outlined above.

Figure 9 shows a schematic representation of CAD data sharing between manufacturers. In this transaction, the part features are essential. The part manufacturing features determine the processes and the machine capabilities required for manufacturing the given part. Currently, such manufacturing features are analyzed using human logic. The framework given in this paper helps to store the CAD data in a new and simple schema where such manufacturing features can be accessed directly. Further, using advanced feature recognition algorithms the features can be labeled. As an extension for this task, in the later stages, the selection of the manufacturing process can be automated using the recognized features.

Conclusion

The proposed model provides the framework for data extraction and storage of part design and manufacturing capability data. The model helps in designing, creating, and populating a database for data generated within manufacturing industries. Moreover, the proposed model can be treated as the framework for understanding the dimensionality of the data for use in predictive analytics. Currently, the

part data are stored in 2d and 3d CAD models. The algorithm expands the storage methods into numerical and machine learnable formats. Part designs mainly explore the geometrical relevance of the feature. Considering this the XML file format uses the geometrical-feature logic for storing the part dimensions. This helps in preserving the geometrical information of the parts. Moreover, the method also considers assemblies and provides single document storage for assemblies with parting into the components using XML tags. This has major application potential for complex assembly and part retrieval processes. The XML storage of design data can be used for matching similar faces between the different parts. This can give rise to an index for understanding geometric symmetry between parts.

The proposed algorithm for dimensional data extraction serves a vast range of applications. Each part that is stored in 3D format can be stored and used into a dimension-based numeric XML file. The part dimensions can be used for creating search engines and further storing the designs into machine recognizable format. The 3D CAD format stores the part design in a 3-dimensional model and often these files are not readable by CAPP tools. An XML file with feature-based dimensions gives a dimensional representation of the part data based on the features. The proposed algorithm reads and extracts the dimensions with respect to each face of the given part design. Such strategy preserves the dimensional and face relevance.

Declarations

Funding: The University of Oklahoma

Conflict of interest/Competing interests: Not applicable

Availability of data and material: Not applicable

Code availability: Available on GitHub: https://github.com/Bhairavsingh/STEP-to-XML/blob/main/STEP_Compelete_Dimensional_Data_Extraction_Rev.04.ipynb

References

[1] M. Alemanni, F. Destefanis, and E. Vezzetti, "Model-based definition design in the product lifecycle management scenario," *Int. J. Adv. Manuf. Technol.*, vol. 52, no. 1–4, pp. 1–14, 2011, doi: 10.1007/s00170-010-2699-y.

[2] F. Z. Fang, Z. Li, A. Arokiam, and T. Gorman, "Closed Loop PMI Driven Dimensional Quality Lifecycle Management Approach for Smart Manufacturing System," *Procedia CIRP*, vol. 56, pp. 614–619, 2016, doi: 10.1016/j.procir.2016.10.121.

[3] W. J. C. Verhagen, *Concurrent Engineering in the 21st Century*. 2015.

[4] D. Sreeramulu and C. Rao, "A New Methodology for Recognition of Milling Features from STEP File," *Int. J. Appl. …*, vol. 6, pp. 172–190, 2008, [Online]. Available:

http://www.publishing.waldenu.edu/ijamt/vol6/iss3/9/.

[5] M. Al-wswasi and A. Ivanov, “A novel and smart interactive feature recognition system for rotational parts using a STEP file,” *Int. J. Adv. Manuf. Technol.*, vol. 104, no. 1–4, pp. 261–284, 2019, doi: 10.1007/s00170-019-03849-1.

[6] R. A. Jaider Oussama, Elmesbahi Abdelilah, “Manufacturing Computer Aided Process Planning For Rotational Parts. Part 1: Automatic Feature Recognition From STEP AP203 Ed2,” *J. Eng. Res. Appl. www.ijera.com*, vol. 4, no. 5, pp. 2248–9622, 2014, [Online]. Available: www.ijera.com.

[7] N. Ismail, C. F. Tan, S. V Wong, M. R. Osman, and S. Sulaiman, “Ruled-Based Feature Extraction and Recognition,” pp. 3–6, 2002.

[8] S. Kalpakjian and S. Steven, *Manufacturing engineering and technology / Serope Kalpakjian, Steven R. Schmid.*, Seventh. Pearson Education, Inc., 2014.

[9] P. Sateesh and M. P, “Subject: Mechanical Engineering IJRIME A METHODOLOGY FOR FEATURE EXTRACTION AND RECOGNITION FOR CAD / CAM INTEGRATION USING STEP FILE ™© all copyrights reserved,” *Int. J. Res. Innov. Subj.*, vol. 4, no. 1, pp. 711–725, 2017.

[10] ScienceDirect, “Eucledian Distance,” 2020. https://www.sciencedirect.com/topics/mathematics/euclidean-distance (accessed Jun. 30, 2020).

[11] S. Abiteboul, P. Buneman, and D. Suciu, “Data on the Web: Semistructured Data and XML,” 2014.

[12] B. Starly, P. Cohen, and S. Raman, “Automating the search and discovery of manufacturing service providers to enable a digital supply chain network,” *Smart Sustain. Manuf. Syst.*, vol. 4, no. 3, 2020, doi: 10.1520/SSMS20200061.

[13] C. Kathe, “IMPROVING PRODUCT DESIGN TOLERANCES USING METR-ONTOLOGY,” University of Oklahoma, 2018.

[14] S. A. IRANI, H.-Y. KOO, and S. RAMAN, “Feature-based operation sequence generation in CAPP,” *Int. J. Prod. Res.*, vol. 33, no. 1, pp. 17–39, Jan. 1995, doi: 10.1080/00207549508930135.

Figures

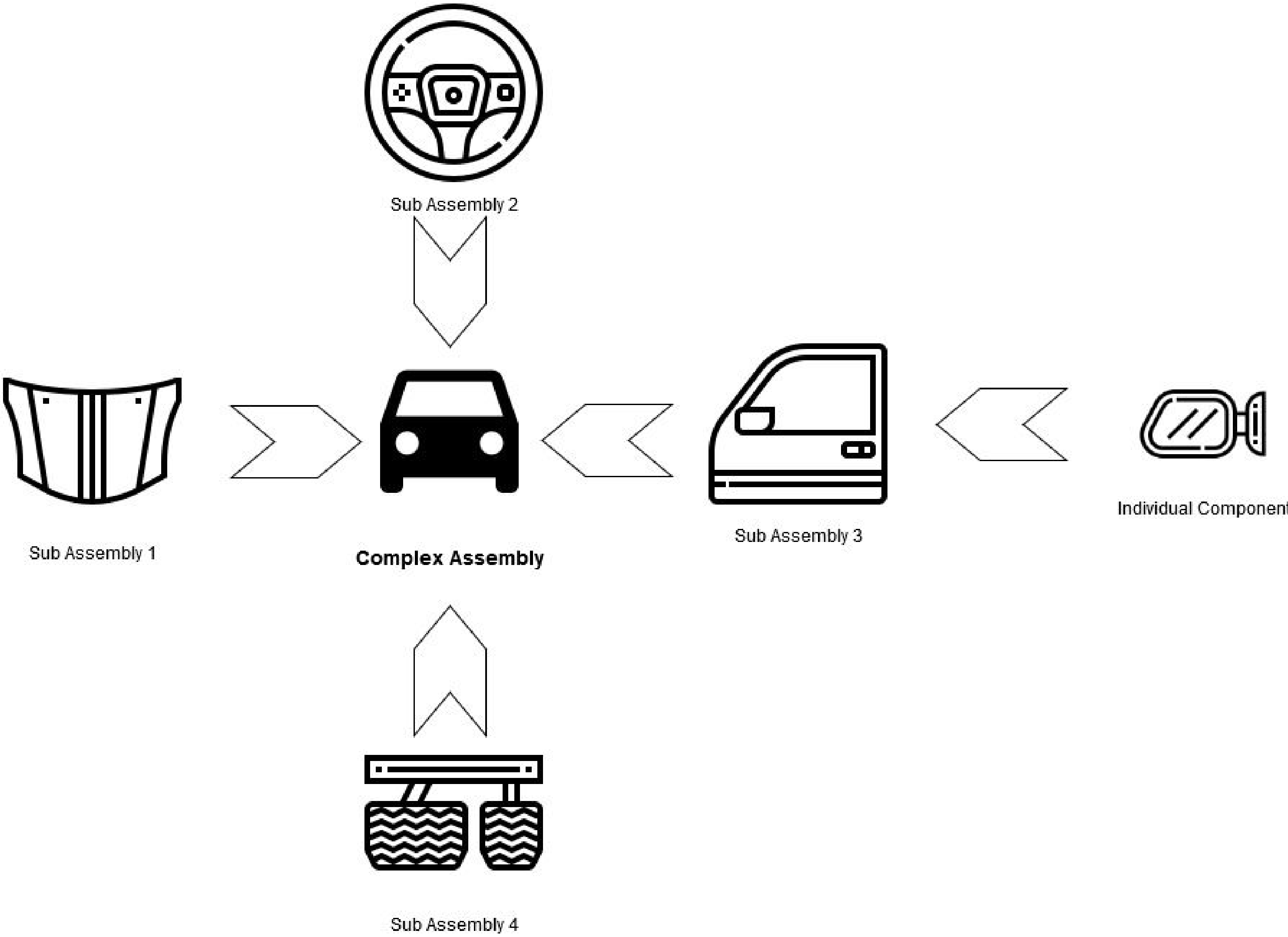

Figure 1

Multi-Level Collaborative Manufacturing

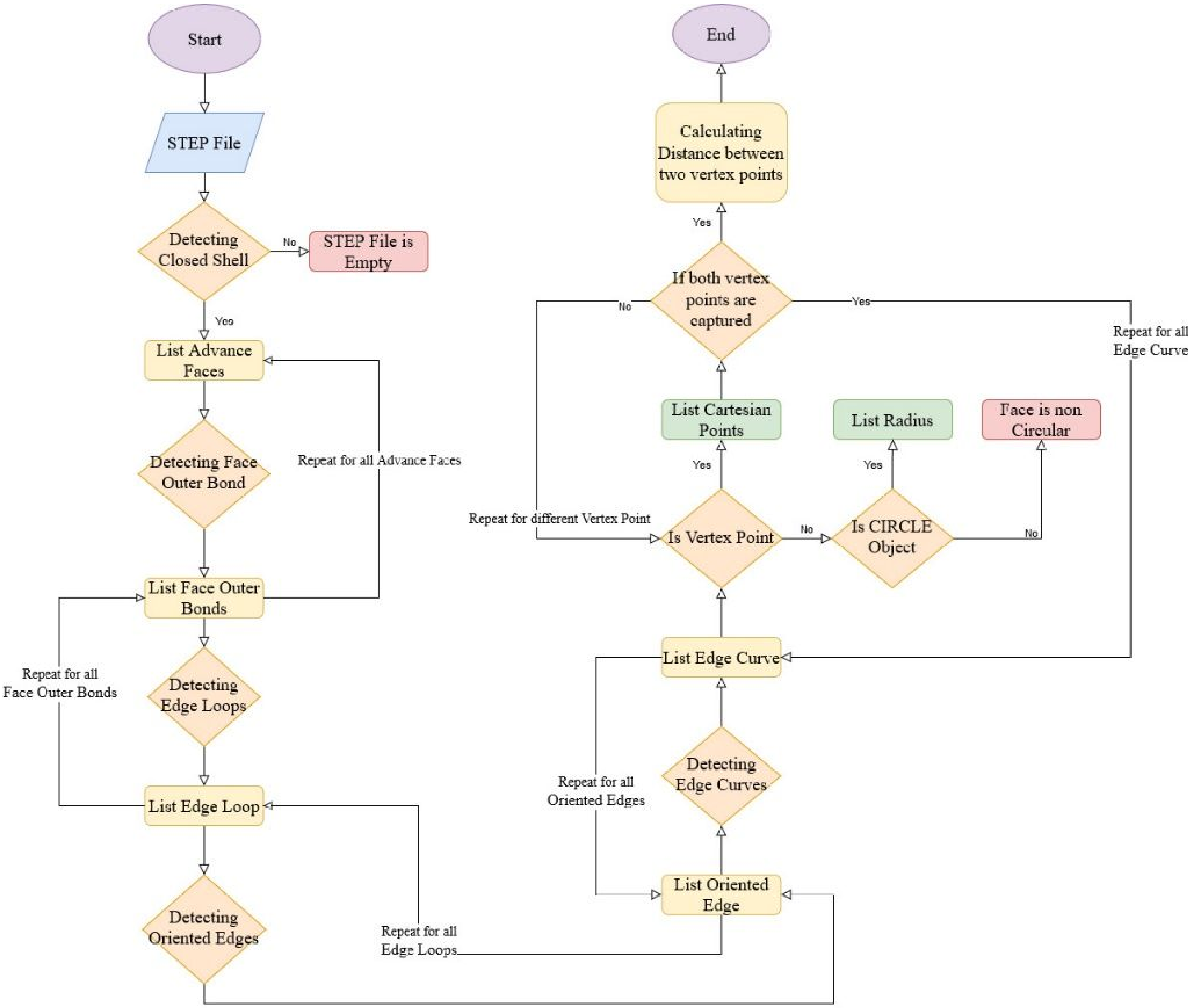

Figure 2

Flowchart of Data Extraction Algorithm

```
<Assembly_Name: __________>
        <Component_1:>
                <Feature_1:>
                        <Dimensions_Circular Radius = _____ />
                        <Dimensions_Linear Length = _____ />
                </Feature_1:>


                <Feature_2:>
                        <Dimensions_Circular Radius = _____ />
                        <Dimensions_Linear Length = _____ />
                </Feature_2:>
        </Component_1:>

        <Component_2:>
                <Feature_1:>
                        <Dimensions_Circular Radius = _____ />
                        <Dimensions_Linear Length = _____ />
                </Feature_1:>


                <Feature_2:>
                        <Dimensions_Circular Radius = _____ />
                        <Dimensions_Linear Length = _____ />
                </Feature_2:>
        </Component_2:>

</Assembly_Name: __________>
```

Figure 3

Snapshot of XML Storage Schema

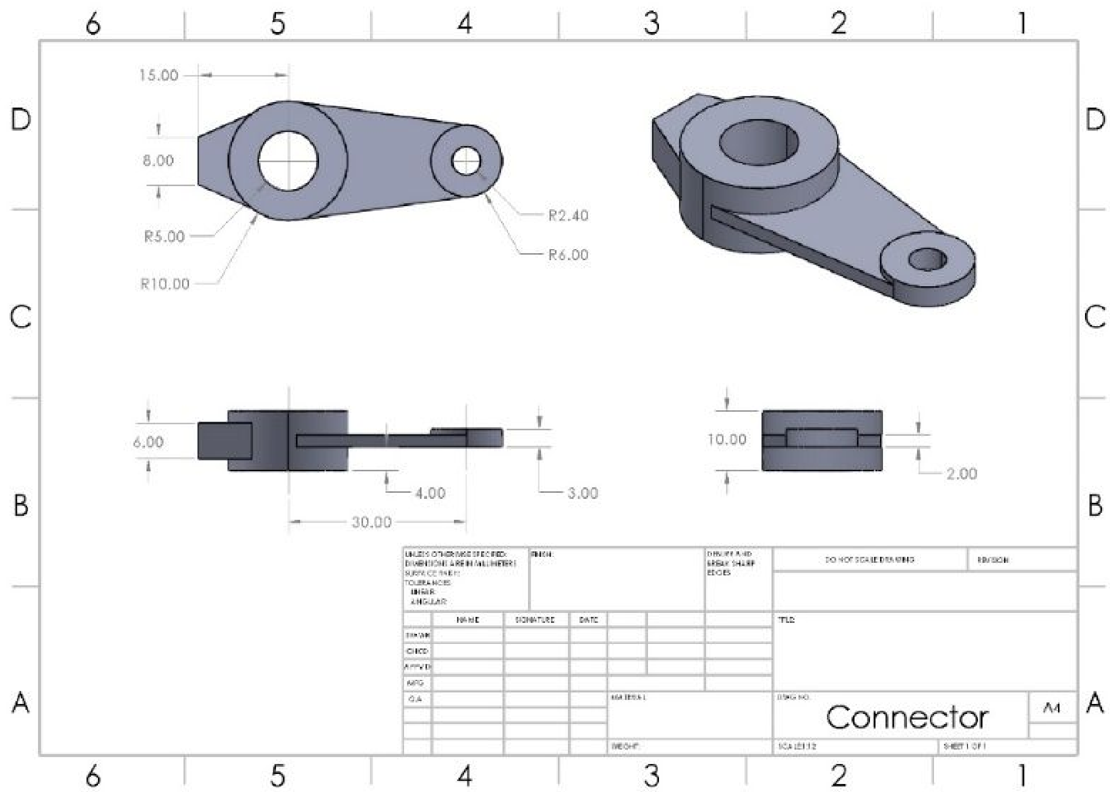

Figure 4

Connector Part Design

```
---------------------------------------------------------------------------
Face Outer Bound:  315
----------------------

        Edge Loop:  328
        Oriented Edges:  ['429', '533', '643', '591']

                Oriented Edge:  429
                Edge Curves:  ['238']

                        Edge Curve:  238
                                Radius:  ['2.4000000000000355']
                                Distance is:  4.800000000000001

                Oriented Edge:  533
                Edge Curves:  ['71']

                        Edge Curve:  71
                                Distance is:  3.000000000000001

                Oriented Edge:  643
                Edge Curves:  ['133']

                        Edge Curve:  133
                                Radius:  ['2.4000000000000355']
                                Distance is:  4.800000000000001

                Oriented Edge:  591
                Edge Curves:  ['424']

                        Edge Curve:  424
                                Distance is:  3.000000000000001
```

Figure 5

Output of Dimensional Data Extraction Algorithm for Connector Part

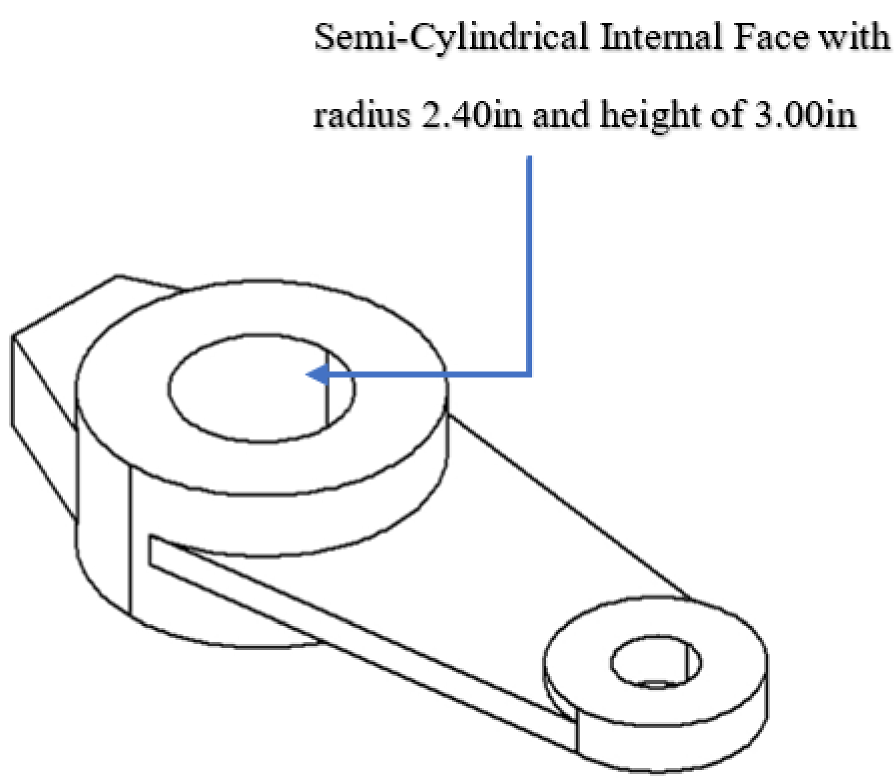

Figure 6

Internal Face of Connector Part

```
  <?xml version="1.0"?>
- <Component_1>
  - <Feature_315>
        <Dimensions_Circular Radius="2.400000000000000355"/>
        <Dimensions_Linear Length="3.000000000000001"/>
        <Dimensions_Circular Radius="2.400000000000000355"/>
        <Dimensions_Linear Length="3.000000000000001"/>
    </Feature_315>
  - <Feature_209>
        <Dimensions_Circular Radius="6.000000000000000888"/>
        <Dimensions_Linear Length="1.0000000000000018"/>
        <Dimensions_Linear Length="1.9999999999999991"/>
        <Dimensions_Circular Radius="6.000000000000000888"/>
        <Dimensions_Linear Length="1.9999999999999991"/>
        <Dimensions_Linear Length="1.0000000000000018"/>
    </Feature_209>
  - <Feature_93>
        <Dimensions_Circular Radius="5.000000000000000000"/>
        <Dimensions_Linear Length="10.0"/>
        <Dimensions_Circular Radius="5.000000000000000000"/>
        <Dimensions_Linear Length="10.0"/>
    </Feature_93>
  - <Feature_191>
        <Dimensions_Linear Length="1.9999999999999991"/>
        <Dimensions_Circular Radius="10.00000000000000000"/>
        <Dimensions_Linear Length="1.9999999999999991"/>
        <Dimensions_Circular Radius="10.00000000000000000"/>
    </Feature_191>
  - <Feature_239>
        <Dimensions_Linear Length="10.0"/>
        <Dimensions_Circular Radius="10.00000000000000000"/>
        <Dimensions_Linear Length="10.0"/>
        <Dimensions_Circular Radius="10.00000000000000000"/>
    </Feature_239>

  - <Feature_43>
        <Dimensions_Circular Radius="5.000000000000000000"/>
        <Dimensions_Circular Radius="5.000000000000000000"/>
    </Feature_43>
  - <Feature_623>
        <Dimensions_Circular Radius="5.000000000000000000"/>
        <Dimensions_Circular Radius="5.000000000000000000"/>
    </Feature_623>
  - <Feature_548>
        <Dimensions_Linear Length="10.0"/>
        <Dimensions_Circular Radius="5.000000000000000000"/>
        <Dimensions_Linear Length="10.0"/>
        <Dimensions_Circular Radius="5.000000000000000000"/>
    </Feature_548>
  - <Feature_563>
        <Dimensions_Linear Length="28.913664589601918"/>
        <Dimensions_Linear Length="1.9999999999999991"/>
        <Dimensions_Linear Length="28.91366458960192"/>
        <Dimensions_Linear Length="1.9999999999999991"/>
    </Feature_563>
  - <Feature_277>
        <Dimensions_Linear Length="28.91366458960192"/>
        <Dimensions_Linear Length="1.9999999999999991"/>
        <Dimensions_Linear Length="28.91366458960192"/>
        <Dimensions_Linear Length="1.9999999999999991"/>
    </Feature_277>
  - <Feature_611>
        <Dimensions_Circular Radius="6.000000000000000888"/>
        <Dimensions_Linear Length="28.91366458960192"/>
        <Dimensions_Circular Radius="10.00000000000000000"/>
        <Dimensions_Linear Length="28.91366458960192"/>
    </Feature_611>
  - <Feature_64>
        <Dimensions_Circular Radius="2.400000000000000355"/>
        <Dimensions_Circular Radius="2.400000000000000355"/>
    </Feature_64>
```


Figure 7

XML File Output of Connector Part

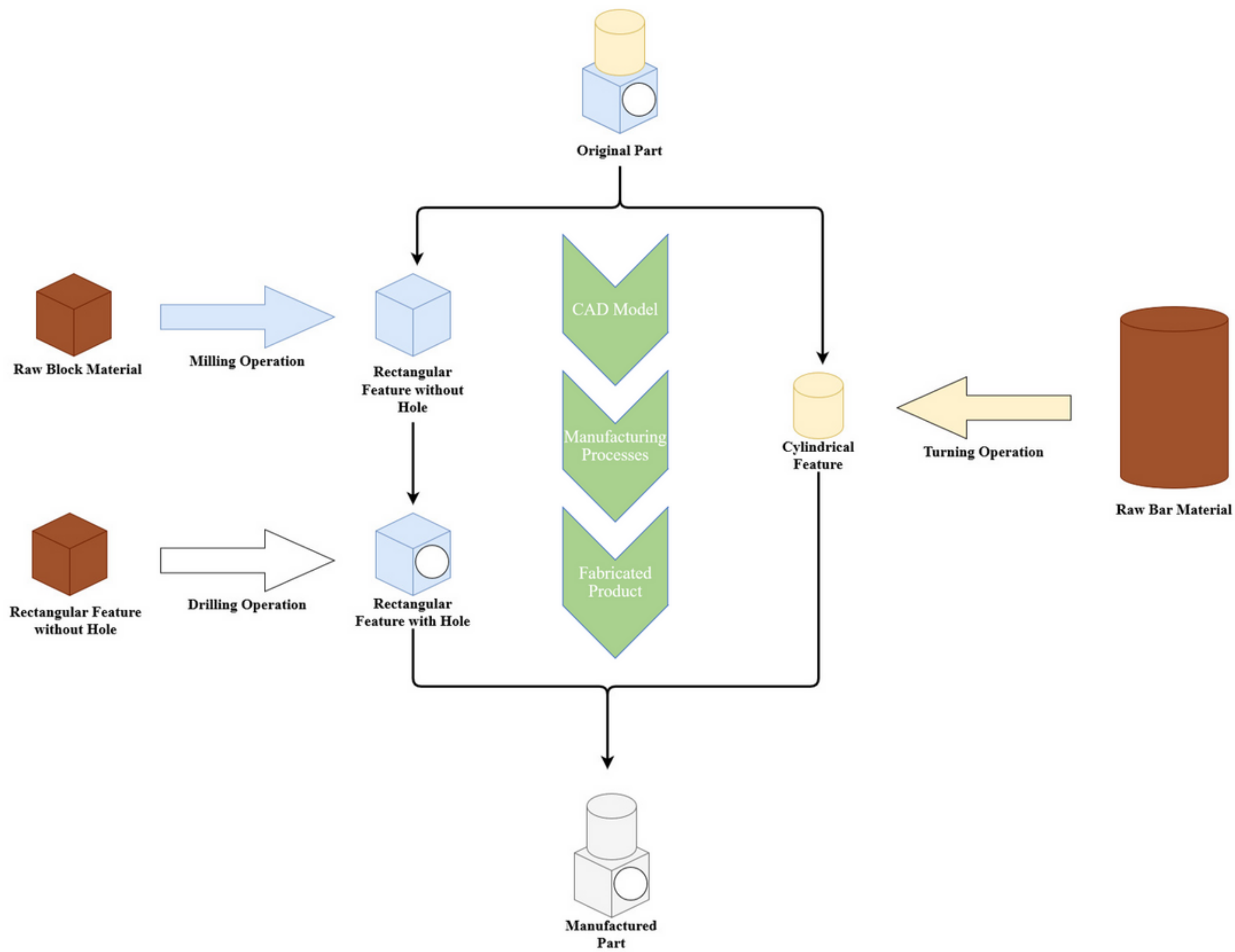

Figure 8

Communication of Design Data in Manufacturing Democratization Chart

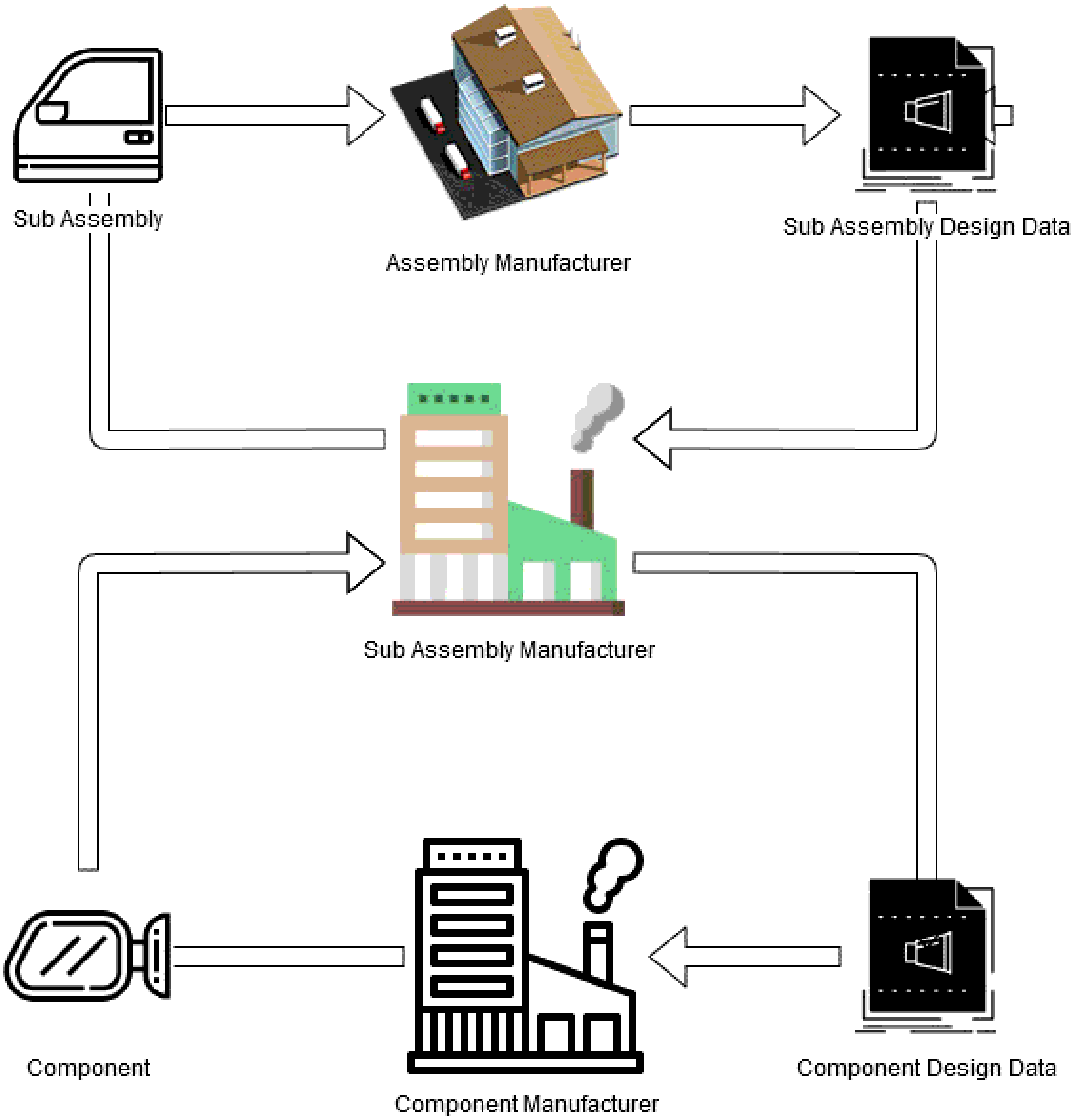

Figure 9

Data Sharing in Collaborative Manufacturing

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- formula.docx