A CAD system for extraction of mating features in an assembly

Arun Mathew and C.S.P. Rao
Department of Mechanical Engineering, National Institute of Technology, Warangal, India

Abstract

Purpose – The information and knowledge about a product and its assembly are necessary to generate all feasible assembly sequences of that product. Assemblies contain a very large amount of information and complex relationships. Identifying assembled parts as well as their contact surfaces is very important in design and manufacturing since this information is essential. The problem is to not only make the information available but also use the relevant information for making decisions, especially determination of the optimum assembly sequence. This paper aims to address these issues.

Design/methodology/approach – This paper describes a system for processing assembly models and extracting assembly related data using application programming interface (API) of the computer-aided design (CAD) software. These data are used to identify the relationships between different components of an assembly thus encouraging generation of feasible assembly sequences.

Findings – Instead of total human interpretation of the assembly design, a direct CAD database interface approach has been proposed to extract the relation with minimal manual involvement. The information extracted is used to generate a list describing the links between the assembled parts, the involved features and the type of link explicitly to facilitate assembly analysis and planning.

Originality/value – The methodology of using the API of the CAD modeling package *SolidWorks*, is a novel approach in which the assembly mate information is captured. Instead of total human interpretation of the assembly design, a direct CAD database interface approach has been proposed to extract the relation with minimal manual involvement.

Keywords Assembly, Modelling, Computer aided design, Data analysis

Paper type Research paper

1. Introduction

Assembly is one of the most important activities in the manufacture of a product because of its complex nature. More than 30 per cent of total industrial product labour costs (Nevins and Whitney, 1980) and 50 per cent of product manufacturing cost are attributed to the cost of the assembly (Rembold *et al.*, 1985). The assembly process consists of a number of different stages, such as putting together all the parts and subassemblies of a specified product. Most mechanical products can be assembled in several ways, meaning that different sequences of assembly operations can result in the same final product. Each such sequence implies a different degree of difficulty for the various assembly operations, resulting from different mechanical constraints imposed by the different sequences of operations. Selection of a good sequence of assembly operations is a crucial factor in maximizing the production profitability and has great impact on the assembly line balancing, machine utilization and feasibility of subassembly operations. Numerous assembly sequence generation methods have been developed to systematically explore all the feasible sequences. In addition to the geometric and topological information, the precedence constraints and inter-part relationships are required to study the assembly. But the above information needed is not openly available. In assembly design, when two pars are assembled together, some surfaces of the two parts will come in contact with each other. The contact area results in a relationship in an assembly (De Fazio and Whitney, 1987).

2. Related work

Many researchers have attempted to generate and evaluate the assembly sequences for a product. Bourjault proposed a procedure which obtains all the precedence knowledge about the liaisons of an assembly by answering a set of structured questions based on his proposed liaison model of the assembly. De Fazio and Whitney simplified Bourjault's procedure and reduced the number of questions to be asked to $2n$ against 2^n of Bourjault's (De Fazio *et al.*, 1993). These two methods study the assembly from the point of view of assembling the product. For the representation of the precedence knowledge for an assembly, there have been several methodologies widely used in the past such as set theory, binary matrix, directed graph, establishment conditions and precedence relationships. However, these methods can only represent partial assembly precedence knowledge. The problem with the above methodologies is that they are difficult to use to generate detailed assembly plans automatically and to deal with the coordination and feasibility of various subassemblies efficiently. An assembly can have many different feasible assembly sequences. As it is difficult to represent each sequence individually, it is necessary to design a method to represent all the sequences in an efficient and compact manner. Gottipolu and Ghosh (2003) developed a method for automatically

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0144-5154.htm

Assembly Automation
30/2 (2010) 142–146

[DOI 10.1108/01445151011029772]

This paper is an updated and revised version of an award-winning paper previously presented at Flexible Automation and Intelligent Manufacturing (FAIM) Conference, University of Teesside, Middlesbrough, UK, 6-8 July 2009.

generating assembly sequences from a solid modeller that generates two types of matrices algorithmically. The aim of extracting inter-part relations from an assembly design is to support automatic assembly sequence generation. According to literature, the inter-part relation have been defined in terms of the spatial relationships (Ko and Lee, 1987), degree of freedom (Eng *et al.*, 1999), liaison relations (De Fazio and Whitney, 1988), hierarchy relationship, mating types (Lin and Chang, 1989). In the joint-based method (Kim *et al.*, 2005), the assembly constraints are assigned on the components, but not on the geometric elements of the components. The method generates assembly models from kinematic joint constraints by extracting feasible joint mating features for each mating component, and then generates the assembly configuration for a set of joint constraints. Geometry-based representations capture the surface mating constraints like fit, coplanar, etc. to establish the relations of precedence and feasibility (Sudarsan *et al.*, 2006). The connection-semantics-based assembly tree hierarchy provides a way to consider both geometric information and non-geometric knowledge of the assembly and obtain the degree of freedom of the mating entities (Dong *et al.*, 2007). Product semantic information model which is made of semantic information is structured into a three level semantic abstract, from which the relevant information is retrieved (Hui *et al.*, 2007). All information regarding relationships between parts should be captured and used during the assembly planning process. Instead of total human interpretation of the assembly design, a direct computer-aided design (CAD) database interface approach using API has been proposed to extract the relations with minimal manual involvement.

3. Modelling of assembly

Assembly modelling is an extension of geometric modelling that facilitates that construction, modifications and analysis of complex assemblies. A product can be considered to be an assembly of elementary components. In assembly modelling, a product model is created representing a product consisting of several smaller components. Parts and components are added to an assembly by specifying mating conditions or constraints. Because of these smaller components, the focus in assembly modelling will be not only on these components, but also on the relations between these components. To describe a product, the elementary components and the relationships between them must therefore be defined. A component that cannot be subdivided into smaller components is called a single part. A group of components merged together is called an assembly. Decisions made during the creation of a model can have great impact on the complete life cycle of the product. Assembly problems are generally handled in graph form. The description of the relationship can be generated manually or automatically, if we have the CAD model of the components. The assembly design process along with the top-level breakdown of assembly form the backbone for the generation of assembly information.

4. Generation of assembly relationships

SolidWorks, the commercial CAD system, is used as the main feature-based design environment. The benefit of using *SolidWorks* is that it includes a complete application programming interface (API) with functions that can be called from Visual Basic. In addition, *SolidWorks* shares the same solid modelling engine as Unigraphics and several other CAD systems like the Pro/Engineer and Catia. Together, these CAD systems account for large user and application bases.

The description of the relationships among the features of various parts is required for an assembly component. These features can be classified into assembly features and primal features. It is the primal features that participate in assembly constraints. The assembly module automatically determines which relationship is meant by the user based upon the features involved in the relationship and updates the degrees of freedom accordingly. The primary mating conditions are align, mate, mate entity, align offset, insert, orient, etc. The align condition requires that the axial centre lines of two parts be collinear. The mate condition requires that the two mating faces lie in the same plane with their outward normal opposing each other. The offset condition requires that the two faces lie in parallel planes with their outward normals in the same direction.

The relationships between a pair of parts are specified by the user in terms of their features and the mating conditions between them. The individual parts in an assembly are created before the assembly module is invoked. The assembly modelling module requires information about the relationships between the part features. The information specified for each mating condition includes the ID of the mating feature and the type of mating conditions as shown in Figure 1.

To build a list of all the characteristics of an assembly, the assembly format is developed to store all the characteristics in an assembly as its signature. The method explores the assembly tree in depth. While exploring the assembly hierarchy, it extracts assembly related information for each part as follows:

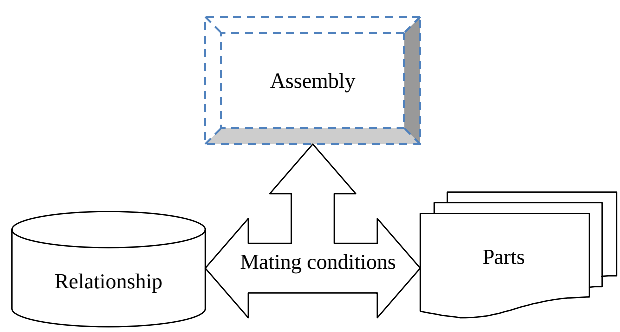
- the method retrieves the constraints and dimensions used to specify the position of the part;
- it identifies which entities are used to constrain the part or subassembly; and
- it identifies the parent features and part of each geometrical entity in use.

The assembly information generated is represented in an object-oriented way to generate assembly strategies.

5. Mate information

The API functions used in this paper are *SolidWorks* functions. The API functions are essential for developing the application software. The names of the mate features, the types, identities and the types of the mate surfaces, the mate clearances and the reference features, etc. are included in the mate information. The names of the mate features, the types, identities and the types of the mate surfaces, the mate

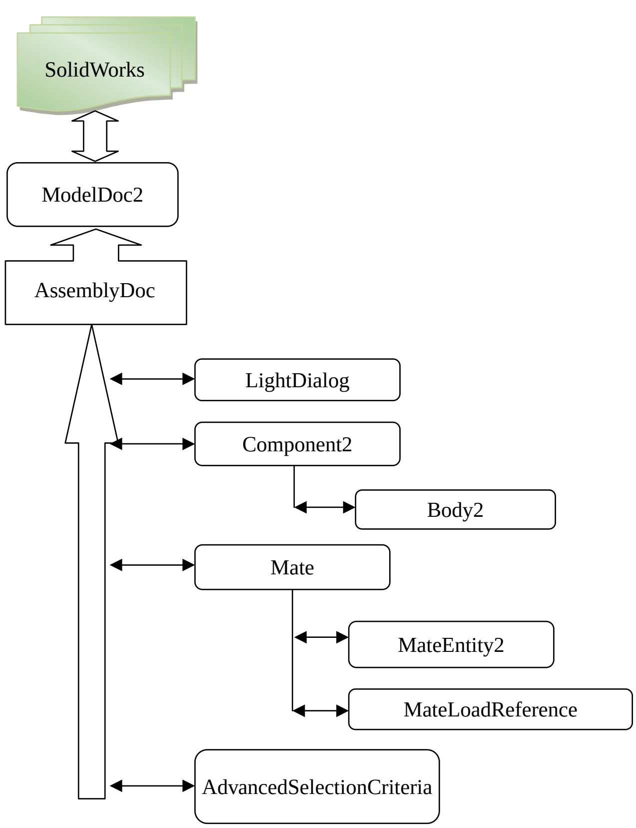
Figure 1 Flow of information



clearances and the reference features, etc. are included in the mate information.

There are three main *SolidWorks* document types namely Parts, Drawings and Assemblies. Each document type has its own object (PartDoc, DrawingDoc and AssemblyDoc) with its own set of related functions as shown in Figure 2 in detail. For example, the AssemblyDoc :: AddComponent4 method exists on the AssemblyDoc object because adding components is specific to assembly documents. The *SolidWorks* API also has functions that are common to all document types. For example, printing, saving, or determining the file name associated with a document would be common operations. To expose common document-level functions, the *SolidWorks* API uses the ModelDoc2 object. The ModelDoc2 object provides direct access to the PartDoc, DrawingDoc, and AssemblyDoc objects. As a general rule, the AssemblyDoc object provides access to functions that perform assembly operations; for example, adding new components, adding mate conditions, hiding and exploding components. The *SolidWorks* API also includes functions that are common to all document types; for example, determining the file name associated with a document is a common operation. To expose common document-level functions, the *SolidWorks* API uses the ModelDoc2 object. The structure of the assembly document is shown in Figure 2. The AssemblyDoc object is derived from the ModelDoc2 object. Therefore, an AssemblyDoc object has access to all of the functions on the ModelDoc2 object. The following objects related to the mate information like mate, mate feature, face and surface. The mate object allows access to various assembly mate parameters. The MateEntity object enables access to mated objects and the assembly mate definition.

Figure 2 Map that shows how to get to object



The feature object allows access to the feature type, name, parameter data and to the next feature in the FeatureManager design tree. The face object allows access to the information of surfaces related to the mate. The surface object provides a function that enables to get the surface type and various surface definition data, as well as, evaluate and reverse evaluation locations on the surface. We gain access to the mate information through these objects in the FeatureManager design tree.

In *SolidWorks*, the current features of assembly bodies are obtained by traversing the FeatureManager tree. An API function of the mate object "GetMateEntities" is used to get the mate entities related to the current mate. *SolidWorks* defines the mate relationships like perpendicular, tangent, coaxial, parallel, distant, angular, symmetric, etc. which can be gained by means of an API function of the feature object – "GetTypeName". The face identities related to the mate entities can be got through an API function of the face object called "GetFaceID".

6. Illustration

To illustrate the process, an example consisting of nine parts of a nut cracker is given. The nut cracker taken for the present study has nine components namely the base, cylinder, hinge, piston, adjuster base, adjuster, connection, handle and pin. *SolidWorks* software was used to model the assembly. The individual components were created as separate geometric models in the part models in the part mode and saved as ".sldprt" files. Next, the assembly modelling mode is invoked and the base is taken as the support component. After specifying the assembly constraints, the assembly was built by adding the remaining components to the base part (BP). Adding to the base component is three sets of components namely the adjuster base, cylinder and the hinge. The adjuster base acts as a subassembly consisting of the adjuster. The cylinder acts as a subassembly having the piston, two connectors and pin as its subcomponents. Further the hinge subassembly consists of the pine and handle which in-turn are connected to the cylinder subassembly via the connectors. All the components are assembled using the mate attributes like the coincident, parallel, perpendicular, tangent, concentric, distance and angle. The completed assembly model is then saved as a ".sldasm" file.

The mate option is used to assemble the faces of two components and the align option is used to align the axis of the two components. Select the faces, edges, planes and so on that is to be mated together. All the mate types are always shown in the PropertyManager, but only the mates that are applicable to the current selections are available. "Coincident" option positions selected faces, edges and planes (in combination with each other or combined with a single vertex) so they share the same infinite plane and positions two vertices so they touch. "Parallel" option places the selected items so they remain a constant distance apart from each other. "Perpendicular" option places the selected items at a 90° angle to each other. "Tangent" option places the selected items tangent to each other (at least, one selection must be a cylindrical, conical or spherical face). "Concentric" option places the selections so that they share the same center line. "Distance" option places the selected items with the specified distance between them. "Angle" option places the selected items at the specified angle to each other. The final assembly resulting from satisfaction of all the mating conditions is shown in Figure 3.

Figure 4 is the assembly's structural view and considering the relevant geometric relations and assembly joints in the assembly, a relation graph can be constructed. Figure 5 shows the state of the assembly showing the base component and the subassembly. A mate graph for the assembly can be constructed as shown in Figure 6. Figure 7 shows the exploded view of the assembly having the nine components. The body base is taken as the base component and the subcomponent consists of the cylinder, hinge, piston, adjuster base, adjuster, connection, handle and pin.

Figure 3 Nut cracker assembly

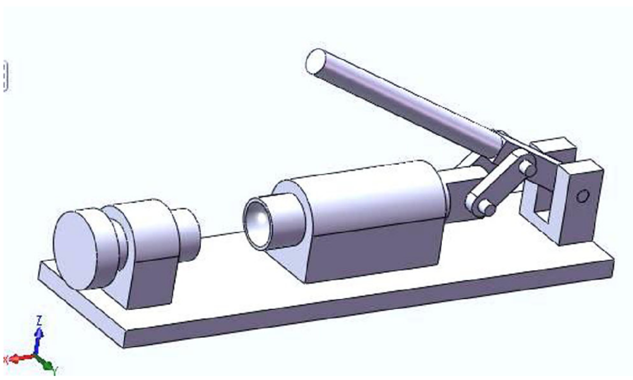


Figure 4 Structure of the assembly

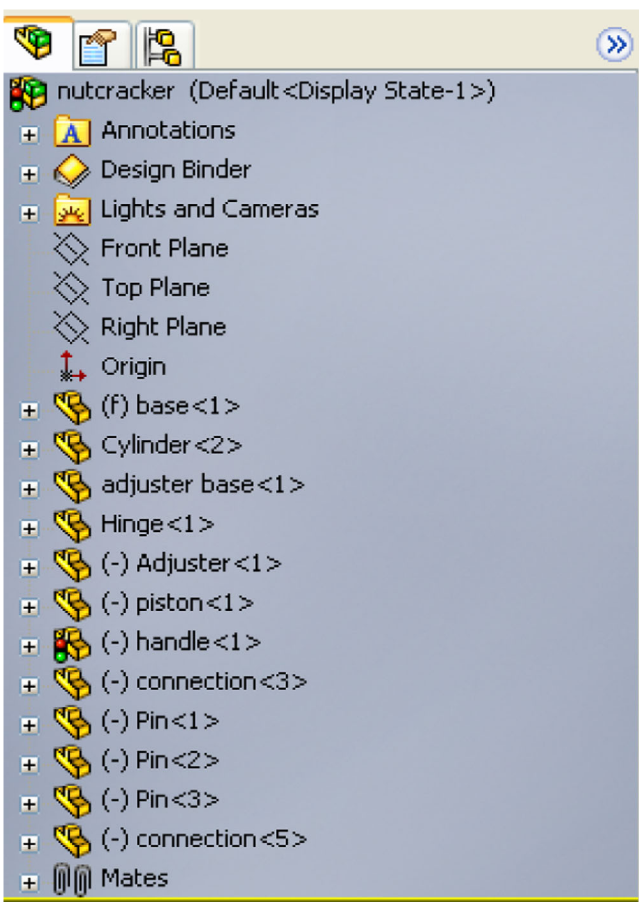


Figure 5 BP and subassemblies

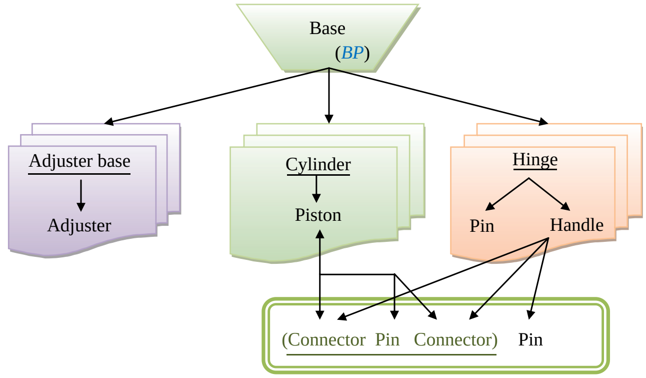


Figure 6 Mate diagram of various connections of the assembly

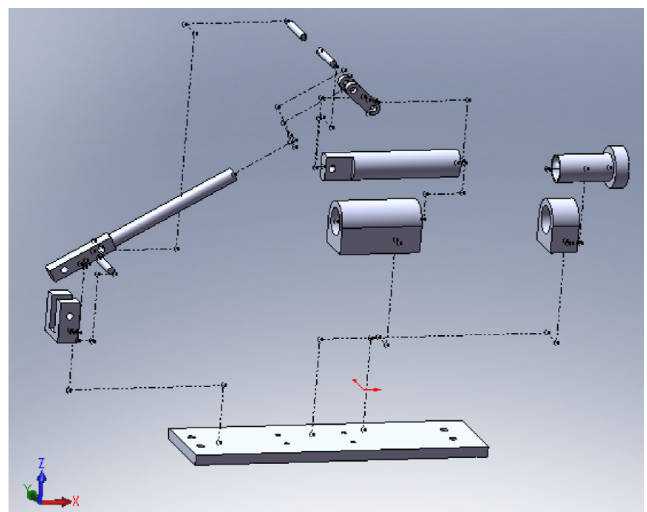
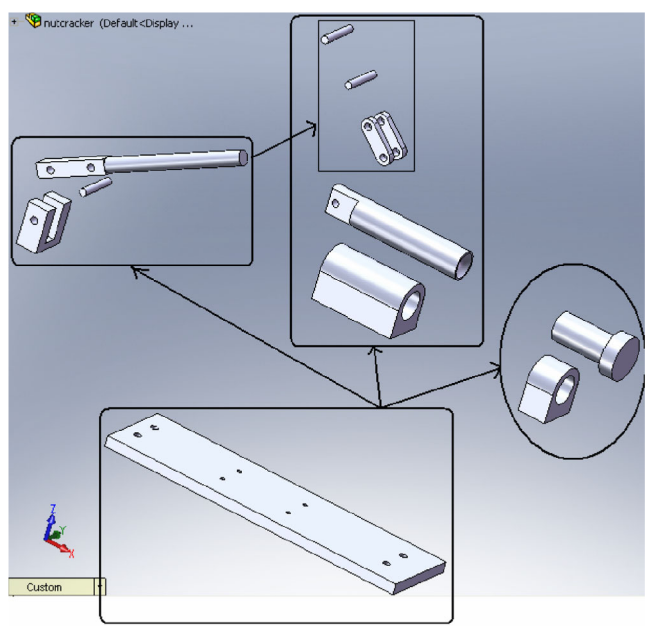


Figure 7 Exploded view of assembly



Assembly Automation
Volume 30 · Number 2 · 2010 · 142–146

7. Conclusion

In this paper, a system for processing assembly models and extracting assembly related data are described. The utilization of mating features for assembly modelling is important, because through these features it is possible to identify the information necessary to perform assembly analysis. The information is used to generate a list describing the links between the assembled parts, the involved features and the type of link explicitly to facilitate assembly analysis and planning. These representations allow the exchange of design intent and assembly constraint information between modelling, analysis and planning systems. The aim of developing such a scheme is to reduce human interaction in the process of creating assembly plans.

References

De Fazio, T.L. and Whitney, D.E. (1987), "Simplified generation of all mechanical assembly sequences", *IEEE Journal of Robotics and Automation*, Vol. 3 No. 6, pp. 640-58.

De Fazio, T.L. and Whitney, D.E. (1988), "Simplified generation of all mechanical assembly sequences", *IEEE Journal of Robotics and Automation Corrections*, Vol. RA-4 No. 6, pp. 705-8.

De Fazio, T.L., Nevins, J.L. and Whitney, D.E. (1993), "A prototype of feature-based design for assembly", *Journal of Mechanical Design*, Vol. 115, pp. 723-34.

Dong, T., Tong, R., Zhang, L. and Dong, J. (2007), "A knowledge based approach to assembly sequence planning", *International Journal of Advanced Manufacturing Technology*, Vol. 32, pp. 1232-44.

Eng, T.H., Ling, Z.-K., Olson, W. and McLean, C. (1999), "Feature based assembly modelling and sequence generation", *Computer and Industrial Engineering*, Vol. 36, pp. 17-33.

Gottipolu, R.B. and Ghosh, K. (2003), "A simplified and efficient representation for evaluation and selection of assembly sequences", *Computers in Industry*, Vol. 50 No. 3, pp. 251-64.

Hui, W., Dong, X., Guanghong, D. and Linxuam, Z. (2007), "Assembly planning based on semantic modelling approach", *Computers in Industry*, Vol. 58, pp. 227-39.

Kim, J.S., Kim, K.S., Lee, J.Y. and Jeong, J.H. (2005), "Generation of assembly models from kinematic constraints", *International Journal of Advanced Manufacturing Technology*, Vol. 26, pp. 131-7.

Ko, H. and Lee, K. (1987), "Automatic assembling procedure generation from mating conditions", *Computer Aided Design*, Vol. 19 No. 1, pp. 3-10.

Lin, A.C. and Chang, H.S. (1989), "A framework for automated mechanical assembly planning", *Journal of Mechanical Working Technology*, Vol. 20, pp. 237-48.

Nevins, J.L. and Whitney, D.E. (1980), "Assembly research", *Automatica*, Vol. 16, pp. 595-613.

Rembold, U., Blume, C. and Dillmann, R. (1985), *Computer Integrated Manufacturing Technology and Systems*, Marcel Dekker, New York, NY.

Sudarsan, R., Han, Y.H., Foufou, S., Feng, S.C., Roy, U., Wang, F., Sriram, R.D. and Lyons, K. (2006), "A model for capturing product assembly information", *Journal of Computing and Information Science in Engineering*, Vol. 6 No. 1, pp. 11-21.

Corresponding author

Arun Mathew can be contacted at: aruntom123@gmail.com