A NEW CAD SCHEMA FOR ASSEMBLY SEQUENCE PLANNING

Arun Tom Mathew [1], Rao C.S.P.[2]

[1]Research Scholar , Department of Mechanical Engineering, National Institute of Technology, Warangal, India,
[2]Department of Mechanical Engineering, National Institute of Technology,Warangal, India
Email: [1]aruntom123@gmail.com

Abstract

The sequence for an assembly design must be exhaustive and validated during the design process quickly to facilitate decisions made during the design process. To meet this requirement, it is essential to develop tools for generating assembly sequences. The effectiveness of an Assembly plan depends more on the way, the assembly is modeled. Using design tools, it’s easy to plan-out the skeleton of the assembly, so that individual components can be designed in the context of the overall structure. As a result, the complexity of managing various relationships between components is simplified. This paper presents a method, to assist the generation of relationships among parts of the assembly using ProToolkit objects that reflects the Pro Engineer Database. By visiting the geometric objects in a solid model, the Object Handles of all the geometric objects can be acquired. This paper also presents different applications in the Automatic Assembly domain using the geometric modeling system as a basis.

Keywords: CAD Assembly, Sequence Generation, Pro Toolkit.

I. INTRODUCTION

The utilization of integrated product models for planning and control of assembly processes has received significant attention over the years. There is much interest in reducing the cost of assembly activities. One way of achieving this is to improve assembly planning. Due to frequent changes in product design and manufacturing methods, it is desirable to automate and computerize the planning activity. Assembly sequence generation plays an important role in designing and planning the product assembly process. The choice of the assembly sequence in which parts of assembly are put together can drastically affect the efficiency of the assembly process.

Identifying part interdependencies in assemblies and planning assembly process are examples of complex decision making activities. The methodology employed in these activities primarily involves matching the technological requirements to the capabilities of the available assembly processes and equipment.

The method described in this paper extracts the feature based assembly information from CAD models of products to facilitate downstream planning and control applications. The assembly model acts as an object oriented database that incorporates component design model with all related information. The main aim is to develop a new representation scheme of assembly relationships and the assembly process for generating detailed feasible assembly sets and assembly plans for a given product.

II. LITERATURE REVIEW

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 2n of Bourjault’s [1]. 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 [2]. However, these methods can only represent partial assembly precedence knowledge. To represent the set of assembly sequences completely, Homen De Mello and Sanderson proposed an AND/ OR graph representation for all the possible configuration of the assembly and generate the assembly sequences for a product using a decomposition method based on the assumption that disassembly sequence is the reverse of a feasible assembly sequence [3]. But then the resultant graph is very large and the number of nodes grows exponentially as the number of parts increases [4]. In the part ordering method proposed by Lee and Ko, a simple sequence is generated and interference checking ensures that the parts to be assembled do not collide during assembly [5]. A different method proposed by Lin and Chang uses a three-layer strategy and a special tree structure to represent the assembly and to generate a feasible sequence [6]. These methods 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 developed a method for automatically generating assembly sequences from a solid modeler [7]. Based on the assembly model, that method generates two types of matrices are used by an algorithmic procedure that generates feasible assembly sequences. Different methods were described for solving the assembly planning problem, and it can be noticed that in those research works the importance of identifying assembled parts for assembly planning decisions to be made. In this paper, a computer system that performs automatic identification of assembled parts is described.

III. MODELING OF ASSEMBLY

A product can be considered to be an assembly of elementary components. In assembly modeling, a product model is created representing a product consisting of several smaller components. 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.

IV. GENERATION OF ASSEMBLY INFORMATION

A. Information

The mate information includes the following: the names of mate features, the types of the mate features, the names of the mate surfaces, the identities of the mate surfaces, the type of the mate surfaces, the mate clearances, and the reference features etc. In Pro Engineer, constraints such as mate, align, insert etc that are used while assembling a component can be got by accessing the database using the Pro Toolkit application.

B.Toolkit

Pro/Toolkit can be described as a software tool through which details and information of any Pro/Engineer application can be retrieved and output in a controlled and customized way depending on the needs of the user. Pro/Toolkit is a C-language customization toolkit for Pro/Engineer uses an Object –Oriented style and provides a large library of C functions that enables the external application to access the Pro/Engineer database and user interface, then integrating any resulting application into Pro/Engineer. The customized ProToolkit library functions are used to retrieve information and data from the design model. A Pro/Toolkit object is a well defined and self contained C structure used to perform actions relevant to that object. Most objects are items in the ProEngineer Database. Pro/Toolkit objects have a hierarchical relationship that reflects the Pro/Engineer database as shown in Figure 1.

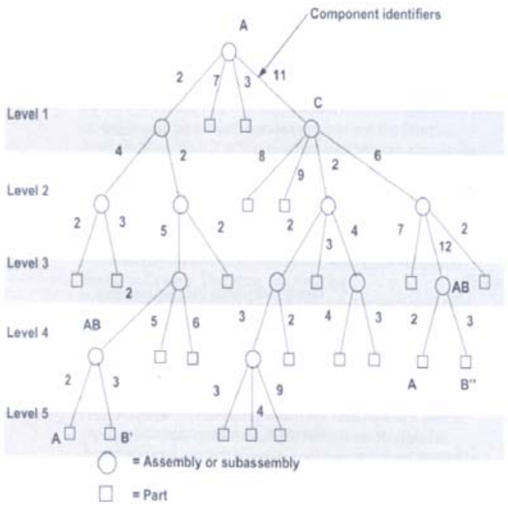


Fig. 1. Sample Assembly Hierarchy

C. Structure of Assemblies and Assembly Objects

The object ProAssembly is an instance of ProSolid and shares the same declaration. The ProAssembly object can therefore be used as input to any of the ProSolid and ProMdl functions applicable to assemblies. In particular, because you can use the function ProSolidFeatVisit( ) to traverse features, you can extract the assembly datum features and their geometry in the same way as for parts. The solid geometry of an assembly is contained entirely in its components. Each components, is a feature of type PRO•FEAT•COMPONENT, which is a reference to a part or another assembly, in addition a set of parametric constraints for determining its geometric location within the parent assembly. ASSEMBLY features that are solid, such as holes and slots, and therefore affect the solid geometry of parts in the assembly hierarchy, do not themselves contain the geometry items that describe those modifications. These items are always contained in the parts whose geometry is modified, within local features created for that purpose of type PRO•FEAT•ASSEM•CUT .

The most important ProToolKit functions for assemblies are those that operate on the components of an assembly. The object ProAsmcomp, which is an instance of ProFeature and shares its DHandle declaration, is defined for that purpose. Each assembly components is treated as a variety of feature, and the integer identifier of the component is also the feature identifier.

An assembly component can be another assembly. In general, therefore, an assembly can contain a hierarchy of assemblies and parts at many levels, in which some assemblies and parts may appear more than once. To identify the role of any database item in the context of the root assembly, it is not enough to have the integer identifier of the item and the handle to its owning pat or assembly, as would be preferred by its ProFeature or ProGeomitem description. It is also necessary to give the full path of the assembly component references down from the root assembly to the part or assembly that owns the database item. The object ProAsmcomppath, which is used as the input to Pro/Toolkit assembly functions, accomplishes this purpose.

The data structure fields are as follows:

Owner – identifies the root assembly

Comp•id•table – An integer array that contains the identifiers of the components that form the path from the root assembly down to the component part or assembly being referred to.

Table•num – Specifies the number of component identifiers in the comp•id•table array.

D. Assembling Components by Feature Creation

In order for the software to perform the identification of parts that are assembled in a product, a data structure is necessary. The modeler has to have hierarchical structure, where a product is composed of an assembly, an assembly can be composed of many subassemblies, a subassembly is composed of the collection of many parts, and each part by one or more features. Each element in this hierarchy has its attributes. However, assembly components are treated as features as features inside Pro/ Engineer, so it is a natural step to replace those dedicated functions by a feature element tree that provides the same functionality, but uses the existing Pro/Toolkit function ProFeatureCreate. This technique offers the following advantages:

i. It conforms to an existing Pro/ Toolkit technique and so does not require special data structures or functions.

ii. It is closer to the Pro/Engineer core code, and is therefore more consistent and supportable.

ELEMENT TREE FOR A COMPONENT ASSEMBLY

PROE FEATURE TREE

- ProE Feature Type
- ProE Component Model
- ProE Component Init Pos
- ProE Component Misc Attr
- ProE Component Constraints

PROE COMPONENT CONSTRAINTS

- ProE Component Constr Type
- ProE Component Comp Constr Ref
- ProE Component Assem Constr Ref
- ProE Component Constr Ref Offset
- ProE Component Constr Attr
- ProE Component User Data
- ProE Component Comp Orient
- ProE Component Assm Orient

V. ILLUSTRATION

To illustrate the process, an example consisting of 9 parts of a nut cracker is given. The nut cracker taken for the present study has 9 components namely the Base, Cylinder, Hinge, Piston, Adjuster base, Adjuster, Connection, Handle and Pin. Pro/ Engineer 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 ".prt" files. Next, the assembly modeling mode is invoked and the body is taken as the base component. After specifying the assembly constraints, the assembly was built by adding the remaining components to the base part. The components are assembled using the assembly features like the mate, align and insert attributes. The completed assembly model is then saved as an ".asm" file.

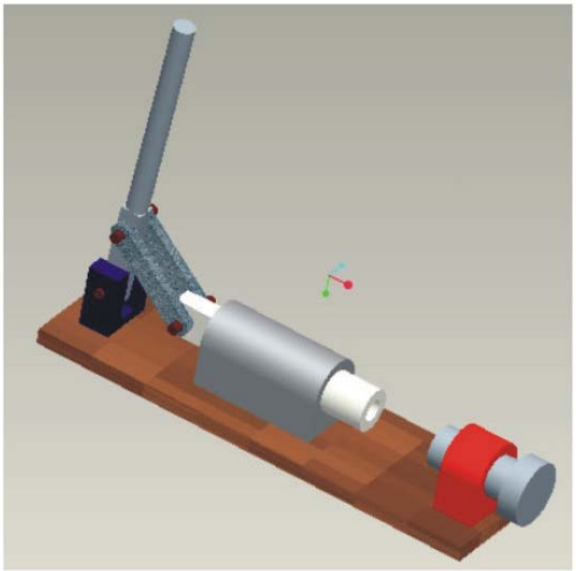
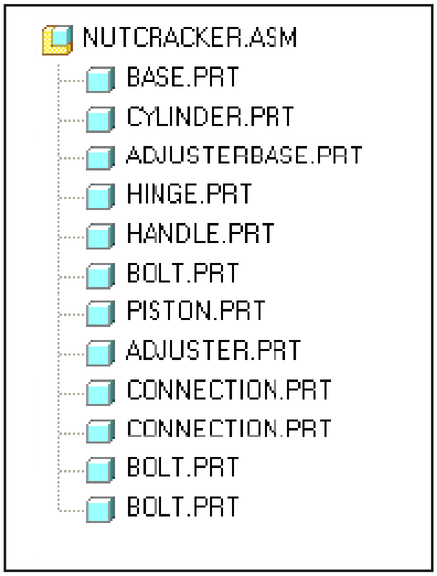


Fig. 2. Nut Cracker Assembly

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. In order to insert a shaft into another the insert option is used in order to seal the assembly. The final assembly resulting from satisfaction of all the mating conditions is shown in the figure 2.



NUTCRACKER.ASM
- BASE.PRT
- CYLINDER.PRT
- ADJUSTERBASE.PRT
- HINGE.PRT
- HANDLE.PRT
- BOLT.PRT
- PISTON.PRT
- ADJUSTER.PRT
- CONNECTION.PRT
- CONNECTION.PRT
- BOLT.PRT
- BOLT.PRT

Fig. 3. Structure of the assembly

Figure 3 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 4 shows the state of the assembly showing the base component and the sub assembly. A mate graph for the assembly can be constructed as shown below.

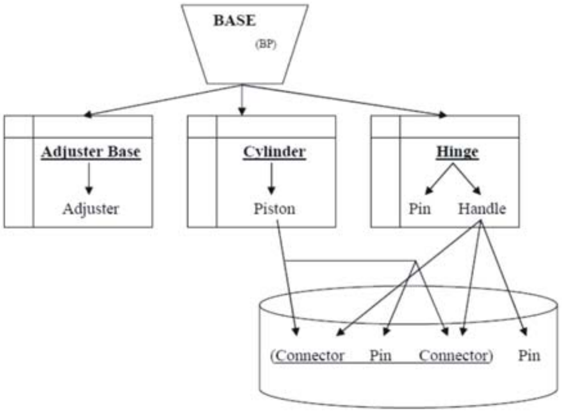


Fig. 4. Base Part and Subassembly

Figure 5 shows the exploded view of the assembly having the 9 components. The Base component is taken as the base component and the subassemblies consists of the Cylinder, Hinge, Piston, Adjuster base, Adjuster, Connection, Handle and Pin.

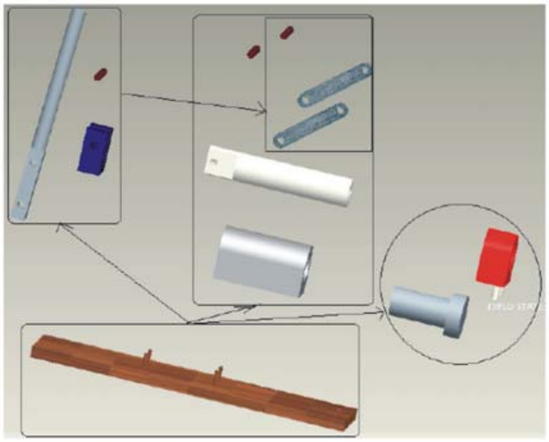


Fig. 5. Exploded view of assembly

VI. CONCLUSION

In this paper, a method to identify assembled parts automatically in a feature based design environment is described. The utilization of features for assembly modeling was important, because through the 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. This list is generated and explicitly represented to facilitate assembly analysis and planning. These representations allow the exchange of design intent and assembly constraint information between modeling, analysis and planning systems.

ProE ToolKit objects have a hierarchical relationship that reflect the Pro Engineer database and visiting the geometric objects in a solid model the object handles of all the geometric objects can be acquired that can be used to access the data in order to generate the information related to the parts in the assembly.

REFERENCES

[1] De Fazio and D.E. Whitney, 1987, "Simplified Generation of all Mechanical Assembly Sequences", IEEE Journal of Robotics and Automation, 3 (6), pp. 640 – 658.

[2] De Fazio TL, 1993, "A prototype of Feature-based design for Assembly", Journal of Mechanical Design, 115, pp. 723–34.

[3] L.S. Homen De Mello and A.C. Sanderson, 1999, "A correct and complete algorithm for the generation of Mechanical Assembly Sequences", IEEE Transactions on Robotics and Automation, 7(2), pp. 228 - 240.

[4] Dini and Santochi, 1992, "Automated sequencing and subassembly detection in Assembly Planning", Annals CIRP, 41(1), pp.1-4.

[5] Ko. H, and Lee K., 1987, "Automatic Assembling procedure Generation from Mating Conditions", Computer Aided Design, 19(1), pp. 3 – 10.

[6] A.C. Lin and T.C. Chang, 1991, "Automated assembling planning for 3-dimensional Mechanical Products", Proceeding of NSF Design and Manufacturing System Conference, pp. 523 – 531.

[7] Gottipolu R.B. and Ghosh K., 2003, "A Simplified and Efficient representation for evaluation and selection of Assembly Sequences", Computers in Industry, 50(3), pp. 251- 264.

Arun Tom Mathew is a researh scholar at the Department of Mechanical Engineering, National Institute of Technology, Warangal, Andhra Pradesh, India. He is a graduate in Mechanical Engineering from Bharathiar University, Coimbatore and a postgraduate in Computer Aided Design from Anna University, Chennai. His main research interests include CAD, Design for Assembly (DFA) and assembly/ disassembly planning and heuristic methods