ISSN 1816-6075 (Print), 1818-0523 (Online)
Journal of System and Management Sciences
Vol. 2 (2012) No. 3, pp. 60-71

3-D assembly process information generation technology based on virtual assembly

Zhang Dan, Zuo Dunwen, Xue Shanliang

College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing, 210016, China

zhangdan430@gmail.com

Abstract: The process of interactive virtual assembly contains mass of assembly process information which is described and saved by assembly hierarchical liaison model. Assembly BOM and 3-D assembly process files generation technology is presented. Data structure of single hierarchical assembly BOM is described, and mapping method of assembly data from assembly hierarchical liaison model to assembly BOM is given. Assembly process files data model is described. The generation flow of 3-D assembly animation based on Pro/Toolkit and assembly process files structure is presented. An application case is given to verify the validation of the proposed technology which can improve the process file editing efficiency.

Keywords: virtual assembly, assembly hierarchical liaison model, assembly bill of material, 3-D assembly process files

1. Introduction

With the development of CAD and Virtual Reality technology, assembly process planning of new product in virtual environment becomes a research hotspot (Jayaram, 1999; Marcelino, 2003; Wang, 2006). Applying virtual assembly technology to complex product's assembly process planning can solve problems caused by traditional assembly process planning method effectively.

Also it can improve the planning efficiency and shorten the design cycle. And results of virtual assembly process planning which includes assembly Bill of Materials data and assembly process files can be used to instruct assembly manufacture in workshop. Traditionally, assembly BOM are obtained by product structure configuration in PDM systems or integration with CAPP systems, and assembly process files are compiled by assembly processors in CAPP systems. There is less research on generating 3-D assembly process information automatically after assembly process planning in virtual reality environment.

By assembly sequence and path planning in the virtual assembly system, not only the assembly process is simulated and validated, but also the product assembly information model is built in the virtual assembly system (Zhang et al. 2010). The model called assembly hierarchical liaison model saves all kinds of process information in interactive virtual assembly effectively. Accordingly this paper discusses the automatic generation technology of assembly BOM and process files based on the assembly hierarchical liaison model. The technology can shorten period of assembly process files compiling greatly.

2. Assembly Hierarchical Liaison Model

The Assembly Hierarchical Liaison Model divides a product into four levels according to complex degree of its assembly relationship: product level, component level, module level and part level. The product level is the top in the model, and it has only one node which denotes the product final assembly. The component level which means collection of sub-assemblies with independent function is lower than product level. Nodes in this level are components of the product, which means a product is composed of these components. The module level which denotes elementary sub-assemblies is higher than the part level, but lower than the component level. Nodes in the module level are units of nodes in the component level, and they are composed of nodes in the part level. The part level is the bottom of the model, and it denotes the minimum unit in whole assembly. The nodes in each level are expressed by directed graph data structure. The vertex and directed edge in the graph denote assembly unit and relationship respectively.

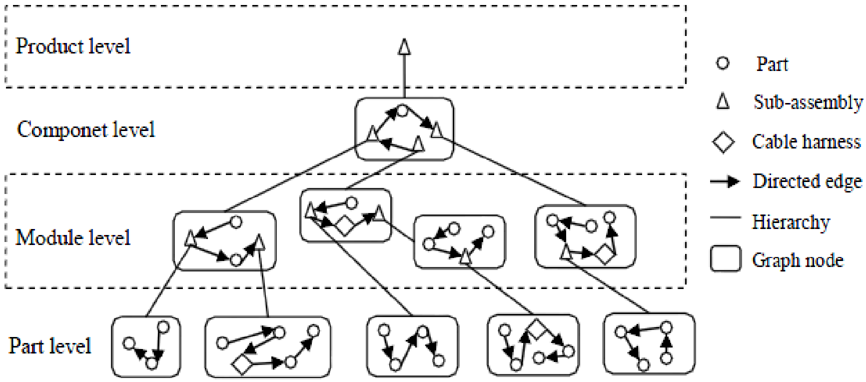


Fig. 1: The schematic diagram of AHLM.

The assembly unit is divided into three types: part, sub-assembly and cable harness. The relationship describes the assembly constraint and sequence, which is shown in Fig. 1.

The data structure of AHLM is described by Backus-Naur Form as follows:

<Graph>::= <GID><Vertexes set><Directed edges set><Parent vertex pointer> <Assembly or Demolition flag> <Status><Level>

<Vertex>::= <VID><Name><Type><Vertex Info. Pointer><Child graph pointer>

<Type>::= <Part>|<Sub-assembly>|<Cable harness>

<Vertex Info.>::= <Position><Attitude><Degree of freedom><Status> <Material><Linked list of assembly path>

<Assembly path >::= <Position><Attitude><Child node pointer>

<Edge>::= <EID><Starting point><End point><Num. of constraints><Constraints set><Num. of tie wrap><Tie wrap set><Process Info. Pointer>

<Constraint>::= <CID><Type><Parameter><ID of first part><ID of bound surface><ID of second part><ID of bound surface><Status>

<Type>::= <Parallel>|<Attach>|<Distance>|<Align>|<Angle>

<Tie wrap>::= <TID><Name><Type><Position><Attitude>

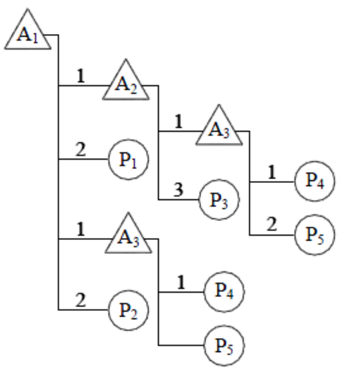
<Process Info.>::= <ID of process equipment><ID of fixture><ID of accessory> <Time><Evaluation index>

<Evaluation index>::= <Assembly time><Num. of assembly people><Movement distance>

3. Generation of Assembly BOM

3.1. Data Structure and Storage of Assembly BOM

Assembly BOM embodies the product structure in assembly stage In PDM systems, and it plays a core data role in production task planning and material management in ERP, MES and other information systems. Assembly BOM needs to express the assembly hierarchy of product, so we use hierarchy structure of BOM view to describe the assembly BOM, which is shown in Fig. 2(a). And the table structure of assembly BOM in database is shown in Fig. 2(b).



(a)

Father ID	Father Version	Unit ID	Unit Version	Num.
root	*NULL*	A_1	001	1
A_1	001	A_2	002	1
A_2	002	A_3	001	1
A_3	001	P_4	001	1
A_3	001	P_5	002	2
A_2	002	P_3	001	3
A_1	001	P_1	001	2
A_1	001	A_3	001	1
A_3	001	P_4	001	1
A_3	001	P_5	002	2
A_1	001	P_2	003	2

(b)

Fig. 2: Data structure and table structure of assembly BOM.

In (a) Assembly A1 is composed of sub-assemblies A2, A3 and parts P1, P2. The sub-assembly A2 is composed of a sub-assembly A3 and a part P3, and the sub-assembly A3 is composed of parts P4 and P5. The number on the edge between two nodes denotes the quantity of the part. The BOM view hierarchy structure expresses father-son relationship and respective quantity relationship among every part in assembly.

3.2. Mapping from AHLM to Assembly BOM

AHLM describes the product structure in assembly stage, so the assembly BOM can be gotten by mapping from AHLM to the BOM view hierarchy structure. The mapping includes two steps: Hierarchy extraction and parts combination. The mapping operation is shown as Fig. 3 in which (a) denotes the AHLM of an assembly A1, (b) is the assembly tree structure and (c) is the BOM view hierarchy structure.

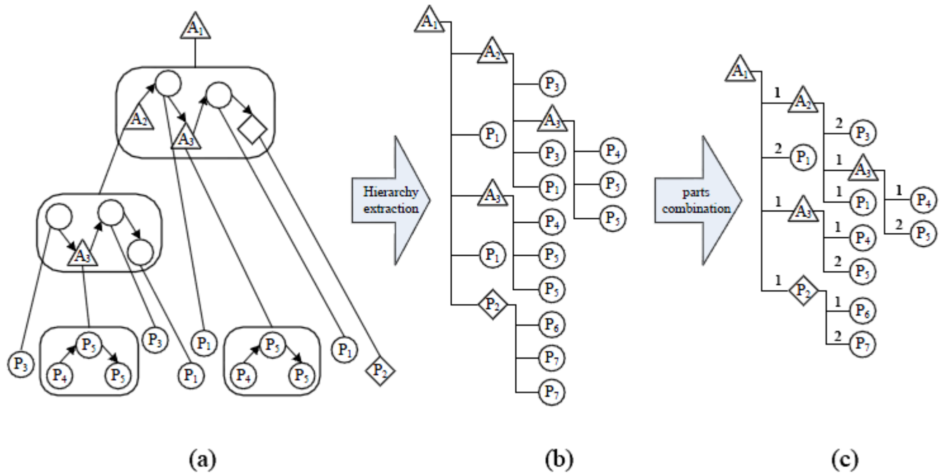


Fig. 3: The mapping operation from AHLM to assembly BOM.

This operation generates an assembly tree after getting the graph data from product level, component level, module level and part level respectively in AHLM.

Hierarchy extraction: First, take the product level node A1 as the root of assembly tree. Second, traverse the vertexes set {A2, P1, A3, P1, P2} in graph node which is pointed by A1 node’s child graph pointer, and according to the sequence of directed edge traverse each vertex as follows: If the vertex is a part, like P1, take it as the child node of root node in assembly tree directly. If the vertex is a cable harness, like P2, take all of the cable branches P6 and electrical connectors P7 as the child nodes of the cable harness, then take the cable harness as the child node of the root. If the vertex is a sub-assembly, like A2, repeat the traversing operation as A1 until all of its child nodes is in part level. Then put the child assembly tree as the child node of the root. The whole assembly tree is shown as Fig. 3(b).

Parts combination: This operation traverses every node whose degree is not zero in assembly tree. Then combines the nodes with a same ID, and records the number of combination. For example, there are two parts P1 in child nodes of A1. After the combination of P1, the number is 2, records it. Then the assembly BOM data structure is generated by combining until the traversing ends.

4. Generation of Assembly Process Files

4.1. Assembly Process Files Data Model

In assembly production, the assembly process files of a product are compiled as procedures and steps. A process volume is defined as files followed by an assembly team in a work station to instruct the continuous assembly work of a product or a component. According to the complexity degree of assembly a process volume is divided to a sequence of procedures, each of which is also divided into a sequence of steps by assembly manipulation. A step corresponding to an assembly manipulation is the least particle size of a process. The entity object model of process volume, assembly procedure and step is shown as Fig. 4.

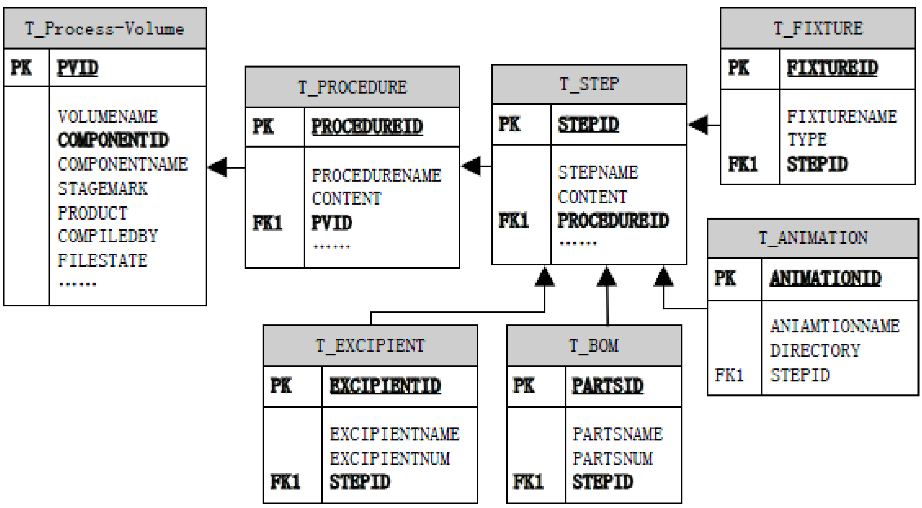


Fig. 4: The entity object model of process volume, assembly procedure and step

A process volume is marked uniquely by PVID, and it has information of component, stage mark, files stage, and etc. A procedure's primary key is PROCEDUREID, and it connects with process volume by foreign key PVID in table T_PROCEDURE. A step's primary key is STEPID, and it includes information of assembly BOM, fixtures, assembly excipients and process animations.

4.2. Generation of 3-D Assembly Process Animation

3-D assembly process animation which can instruct workers intuitively is the important part of an assembly process file. The assembly process information which mainly includes assembly path and assembly sequence can be recorded during product virtual assembly. Data structure of assembly path and sequence is described as follows:

```
typedef struct _Assemblypath{
pos position;                    //position
ori orientation;                 //orientation
assembly_path *pnextpath;        //pointer to next path
} assembly_path
typedef struct _Assemblysequence{
int id;                          //ID
Cnode *part;                     //pointer to assembly model
assembly_path *ppartpath;
assembly_sequ *pnextpart;
} assembly_sequ
```

The 3-D assembly process animation is made by second development interface Pro/Toolkit of Pro/Engineering in ANIMATION module. The animation file generated from the module can be a video file with MPEG format which has advantages as: the compression of images is higher and needs less storage space, thus it's suitable for networks transmission and play. The flow of assembly process animation generation by assembly path and sequence information is shown as Fig. 5.

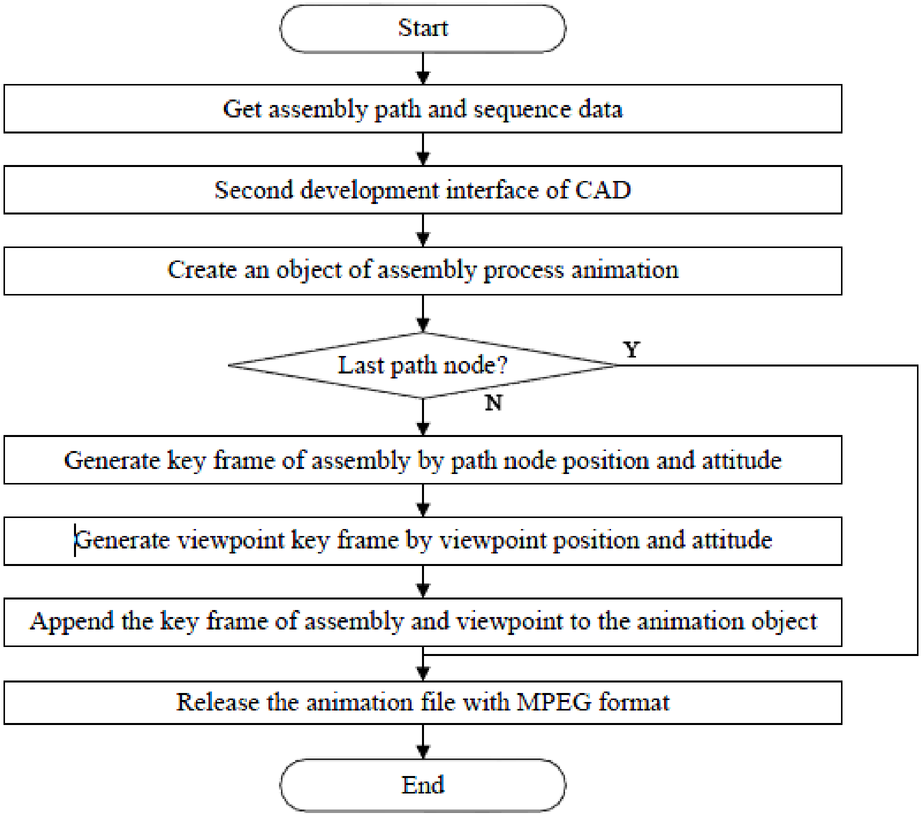


Fig. 5: 3-D assembly process animation generation flow based on Pro/Toolkit.

4.3. Generation of Assembly Process Files Structure

In graph node of AHLM, each vertexes records parts for assembly and their assembly path information. Each directed edge records assembly sequence and process information among parts corresponding to the assembly step content, assembly BOM, fixture and assembly excipient information in assembly process files data model respectively. So we can generate assembly process files structure and their main content.

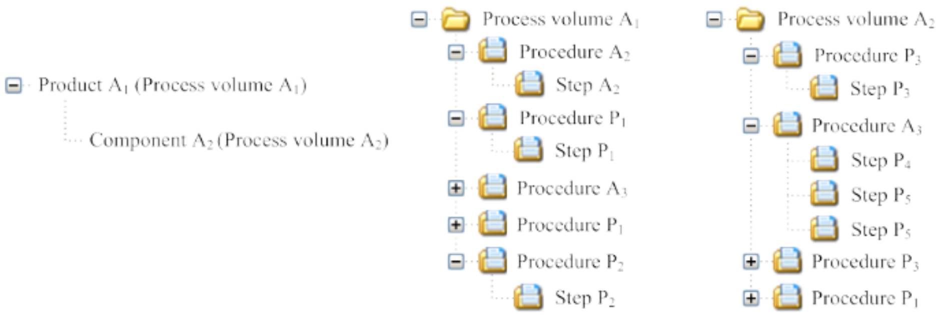
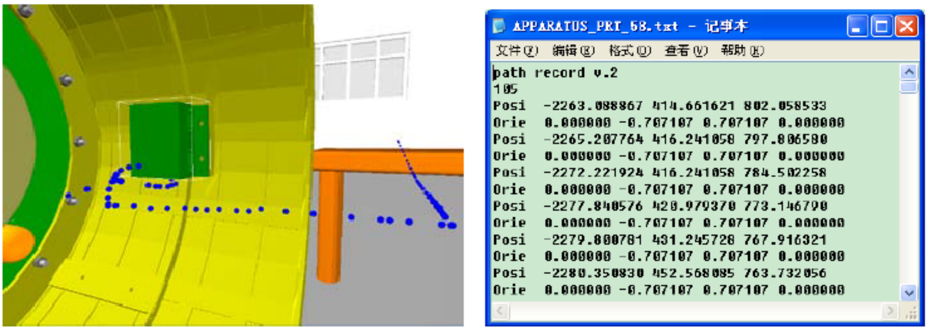


Fig. 5: The generation of process volume catalog tree by assembly tree

The generation of process volume catalog tree by assembly tree can be combined by following steps: First, only preserve the assemblies in produce level and component level, and generate respective catalog tree nodes which is shown in Fig. 5(a). Second, generate the procedure nodes in process volume catalog tree according to the next nodes in assembly tree which is shown in Fig. 5(b). At last, generate the step nodes by decomposing the remaining nodes in assembly tree, and put the information of assembly parts, fixture, excipient in directed edge into assembly step tables in database which is shown in Fig. 5(c).

5. Example

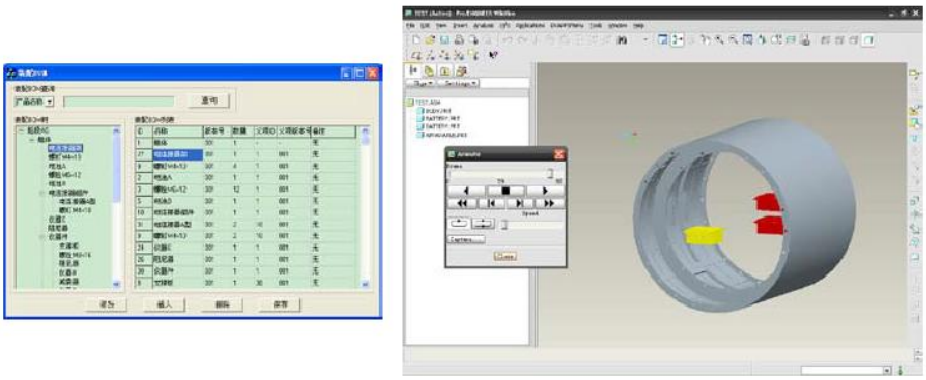
A 3-D assembly process information generation software has been developed by VC++ 6.0 based on a desktop virtual assembly system DVAPPS. An information integration interface with Pro/Engineering Wildfire 4.0 is developed by second development interface Pro/Toolkit. Take a component assembly model of a product for an example to validate the generation technology, which is shown as Fig. 6 and Fig. 7.



(a) (b)

Fig. 6: Assembly path of an instrument part model displayed by a sequence of points in DVAPPS and the storage file of assembly path data

Fig. 6(a) shows assembly path of an instrument part model displayed by a sequence of points in DVAPPS. Fig. 6(b) shows the storage file of assembly path data where each path point is composed of position and attitude data. The attitude is expressed by quaternion.



(a) (b)

Fig. 7: The assembly BOM interface generated according to AHLM

The figure shows the assembly BOM interface generated according to AHLM. On the left of interface shows the tree of assembly BOM and on the right the interface shows the list of assembly BOM which supplies operations like insert, modify, save and delete. Fig. 8 shows the interface of animation generation in Pro/Engineering: Click the sub-menu item “LoadPathFile” on the user-defined pull-down menu “DVAPPSMenu”, it will popup file-selection dialog box.

When you select path files as Fig. 6(b) shows, the system will load the path data by the order of sequence, and automatic call the Pro/Engineer Animation module. Then the animation is made by the module. When click the “Capture…” button on the panel of Animation module, it will remind you to save the assembly process animation file with MPEG format.

6. Conclusion

Virtual assembly technology supplies a novel low cost approach to solve the complex products’ assembly problem, which can take agile and economical assembly process planning. During the planning in virtual environment, AHLM records mass of 3-D assembly process information, so how to fast generate the 3-D assembly process files needs to research. This paper studies on the automatic generation technology of assembly BOM and process files based on the AHLM in virtual assembly system. The mapping operation from AHLM to assembly BOM is presented by expressing data structure of assembly BOM. Assembly process files data model is described. Generation flow of 3-D assembly process animation and assembly process files tree structure is presented. Finally effectiveness of the generation technology is verified by an example. The technology can effectively shorten the assembly process files compiling period to improve development efficiency of new product.

Acknowledgements

This research is supported by Postdoctoral Research Financial Aid Plan of Jiangsu Province (1102053C) and Introduced Talents Research Starting Funds of NUAA (1005-56YAH11035).

References

GUO, G., CHEN, G. B., & LIU, F. (2004). Expression and application of mono-tier and multi-tier BOM within product lifecycle. *Computer Integrated Manufacturing Systems*, 10(1), 59-64.

Jayaram S, Jayaram U, & Wang Y. (1999). VADE. A virtual assembly design environment. *IEEE Computer Graphics and Applications*, 19(6), 44-50.

Marcelino L, Murray N, & Fernando T. (2003). A constraint manager to support virtual maintainability. *Computers & Graphics*, 27 (1), 19-26.

Wang, Q. H., Li, J. R., & Gong, H. Q. (2006). A CAD-linked virtual assembly environment. *International Journal of Product Research*, 44(3), 467-486.

Zhang, D., Zuo, D. W., Jiao, G. M. et al. (2010). Study on mixed assembly modelling of rigid and flexible parts for aerospace product virtual assembly. *Key Engineering Materials*, 432, 9-12.