23

Assembly features and sequence planning

Winfried van Holland and Willem F. Bronsvoort
Delft University of Technology
Faculty of Technical Mathematics and Informatics
P.O. Box 356, NL-2600 AJ Delft, The Netherlands
email:(W.vanHolland/Bronsvoort)@cs.tudelft.nl

Abstract

Feature models are now widely accepted in design and manufacturing, but still hardly used in assembly. We do use assembly features, and show how this can profitably be done in assembly sequence planning. Assembly sequence planning is highly dependent on several other planning tools, e.g. grasp planning, motion planning and stability analysis, to find precedence relations between components. For these planning tools the effectiveness of using features is also shown.

Keywords

Feature modelling, assembly modelling, assembly features, assembly sequence planning, clustering

1 INTRODUCTION

Especially in flexible assembly, where a relatively small number of products must be assembled in a limited period of time, computer support is needed in all kinds of planning. A special kind of planning is assembly sequence planning, special in the sense that it is highly dependent on other planning modules, e.g. grasp planning, motion planning and stability analysis (figure 1).

Often assembly sequence planning is reduced to only investigating a precedence relation graph, to find the optimal assembly sequence. The creation of the precedence relation graph is then left out of consideration. It is a combinatorial problem to find an assembly sequence from the precedence relations, and although the solving technique is not that easy, one can use a common search technique to find a solution.

However, to find the precedence relations, information relevant to manufacturing and assembly is needed. This information is mostly not, or only to a limited extent, present in the product model. The designer can only store some general information in the product model, i.e. the geometry and topology of the product, the position and orientation of the different parts, and sometimes simple relations between these parts. Relevant information known to the designer is lost during modelling, because it could not be stored in the product model. Analysis programs then have to be executed to retrieve similar information from the product model needed for grasp planning, motion planning, stability analysis, etc., and this can be very difficult. It is much easier to store the information in the product model during the design phase (Wilson and Pratt 1988).

M. J. Pratt et al. (eds.), *Product Modeling for Computer Integrated Design and Manufacture*

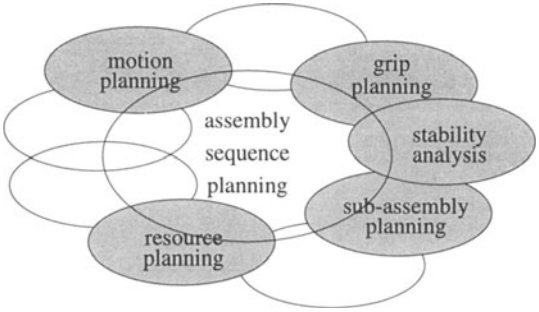


Figure 1 Assembly sequence planning and its relation with some other planning tools

This problem can be solved for a great deal by using feature models (Bronsvoort and Jansen 1993). All relevant information can be stored within a feature model, and can be used by several analysis and planning tools. An additional benefit of using feature models, instead of geometric models, is that these are on a higher abstraction level closer to the way of thinking of designers and engineers.

A new approach in assembly modelling and planning is thus to make use of feature models. Although design and manufacturing features can be used to retrieve some information used in assembly, these features only give information on single parts. We, therefore, extend the feature concept, and store assembly-specific information in so-called assembly features. We distinguish two types: handling and connection features, storing respectively information specific for handling a component and on connections between components (van Holland and Bronsvoort 1995). In this paper, we show how these assembly features can be profitably used in assembly sequence planning.

In section 2, a brief overview will be given on known assembly sequence planning methods. Section 3 will describe assembly features. Section 4 shows how assembly features can be used within assembly sequence planning. In section 5, some conclusions are drawn.

2 ASSEMBLY SEQUENCE PLANNING

In this paper, often the terms parts and components are used. To avoid confusion, the differences between both are explained. A *part* is the smallest entity in a product; it cannot be subdivided into smaller entities. A *component* is a set of parts that is stable, i.e. it does not fall into pieces during assembly. Because a part is always stable, it is also a component.

Assembly sequence planning is often reduced to find the optimal, or a semi-optimal, assembly sequence out of all feasible assembly sequences. Feasible assembly sequences are those assembly sequences for the components in a product that can create the complete product in practice. The optimal or semi-optimal sequence (called optimal sequence for short in this paper), is the sequence with the optimum or a semi-optimum for total assembly time, used resources, or combinations of these properties.

However, the whole process of assembly sequence planning is more comprehensive, and can be subdivided into three main steps, of which sometimes the first two steps are taken together:

1. generate precedence relations between the components in the product,
2. generate all feasible assembly sequences,
3. find the optimal assembly sequence from the feasible sequences.

As was stated in the introduction, to find the precedence relations between components, information relevant to manufacturing and assembly is needed that is usually not available in the product model. Sometimes this is gathered by interrogating a human assembly planner, but mostly computer tools are used. These tools take as input a geometric description of the product, with sometimes simple relations between components. The simple relations are restricted to whether components mate with each other, and whether these matings are fixed or not. Sometimes, the simple relations are combined for use in reasoning applications, but there the same precedence relations must be computed. A recent overview of techniques in assembly sequence planning is given by Gottschlich et al. (1994).

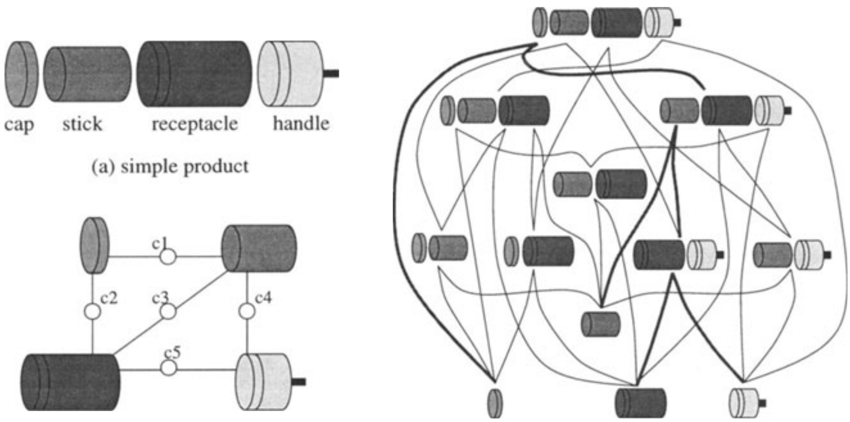
The precedence relations are used to generate all feasible sequences, which can be stored in an *AND/OR* graph (Homem de Mello and Sanderson 1991) or in an Assembly State Transition Diagram, called *ASTD* (Waarts et al. 1992). Mostly the AND/OR graph or ASTD is built simultaneously with investigating the precedence relations. The latter is done by finding feasible disassembly sequences for the product, and reversing them to get the feasible assembly sequences. Every component in the product is investigated in turn whether it can be disassembled from the product. This is done by using precedence relations of the component and the other components in the product. How these precedences can be found is described in the next subsection. If the component can be disassembled, it is removed from the product, and the procedure is repeated on the components left, until all components have been taken out. Finding disassembly sequences is easier than finding assembly sequences, because every step in disassembling a product leads to a smaller product that can always be disassembled, whereas every step in assembling a product leads to a bigger product with increasing chance on obstruction for components still to assemble. The latter can result in checking many infeasible sequences. Figure 2 shows an example product (a) with its relation graph (b) and resulting AND/OR graph (c).

Out of the set of feasible assembly sequences, the optimal sequence must be selected. This can be done by applying some kind of cost function to every feasible sequence, and to take the one with the optimum. An example of such an optimum sequence is shown in figure 2c; it is represented by the thicker lines. Because of a combinatorial explosion, there are many sequences to investigate, and heuristic search methods are used to get the optimal sequence.

Sometimes some of the steps can be done off line, i.e. before the actual assembly process, e.g. to find an optimal assembly sequence for a product that must be assembled on a specific machine. In flexible assembly, however, and especially when parallel assembly with shared resources is considered, on-line planning must be used. A sequence generated on line can be better than a sequence generated off line, because the availability of resources can be taken into account.

Finding precedence relations

How the precedence relations between components can be found is described by giving some examples of used tools, in particular grasp planning, motion planning, and stability analysis. Notice that in assembly sequence planning, these tools are only used to find precedence relations, and not to find all information needed for the assembly process. For example, in grasp planning it is sufficient to determine that a grip can be found, whereas later the exact



(b) relations between components

(c) feasible assembly sequences

Figure 2 Example of feasible sequences for a simple product

grip can be determined. The latter will take more computation time.

Grasp planning is used to find specific grip positions for a gripper to grasp the component. To find these grip positions, first possible grip areas on a component are determined using the geometry of the component. Contact areas between component on the one hand and fixtures, feeder or the already assembled components, called *partial assembly*, on the other hand, are removed from these possible grip areas. A gripper finger cannot be positioned on these areas. These possible grip areas are used to find actual positions for the gripper fingers, giving a stable grip on the component. If such a stable grip cannot be found, there is some precedence relation between component and other components in the partial assembly, because the component cannot be disassembled from the partial assembly.

In motion planning, two steps are considered, gross motion and fine motion planning.

In gross motion planning, a collision free path is searched for a specific component, towards the partial assembly. Mostly a gross motion planner computes a path for a component from the feeding position to the insertion position, a position near the final assembled position where there is still no contact between component and partial assembly. The geometry from component and partial assembly is taken, and an obstacle-free path is calculated in 3D space.

Fine motion planning is used to find a path from the insertion position to the final position. Here the contact areas between component and partial assembly are used to find possible motion directions. Specific connections, e.g. pin-hole, face-face, threaded, etc., often have their own specific fine motion strategies. To select the right strategy, the type of connection must be retrieved from the geometry, which is sometimes very hard, or even impossible.

If no gross motion or fine motion path can be found for a component with respect to a partial assembly, then there is again some precedence relation between the component and other components in the partial assembly, because the component cannot be disassembled from the partial assembly.

Stability analysis is used to find whether the new partial assembly, created after a component has been assembled on it, remains stable. In stability analysis, not only the geometry of the components is needed, but also the relations between components must be taken into account. The latter is often a problem, because very little information on that is stored in product models. The contact areas are needed to find possible motion directions, which in turn are used to check for gravitational stability. In disassembly planning, this means that every time a component is taken from the partial assembly, the new partial assembly must be checked for stability. If it is unstable, then the component cannot be disassembled, and thus there exists some precedence relation between the component and other components in the partial assembly.

Although not all required planning tools have been described above, it does give some notion about which kind of information has to be retrieved from product models. Notice again that this information is mostly already known to the designer or engineer. The main conclusions here are that in the planning tools, information is retrieved from a low-level product model, and that how this information must be retrieved is programmed within the tools.

3 ASSEMBLY FEATURES

Information retrieval using feature models is quite different. There is less need for applications that know how to retrieve specific high-level information out of general low-level information, because much specific high-level information is already present in the product model, stored within features.

Feature modelling

Feature modelling is now widely accepted in design and manufacturing processes, see Bronsvoort and Jansen (1993) for an overview. Before feature models were used in manufacturing, all kinds of applications were developed using low-level geometric models, to find how the product could be manufactured. Storing manufacturing-specific information within standardised form elements in the product model, significantly improved the planning process.

The information that can be stored in features is of course not restricted to manufacturing information, it can also be design-specific or assembly-specific information. We therefore define a *feature* as a physical part of an object mappable to a generic shape and having functional significance.

Feature models can be created by recognising features in the geometric model, called *feature recognition*, or by defining the product with features, called *design by features*. At Delft University of Technology, we are developing a multi-disciplinary feature modeller, called SPIFF (Bronsvoort et al. 1996, de Kraker et al. 1996), using the design-by-features and feature-conversion concepts, to be used in a concurrent-engineering environment.

In assembly, the concept of features has been used only by a few researchers (Shah and Rogers 1993, De Fazio et al. 1993). We enhance the already known feature concept with *assembly features*, defined as features with significance for assembly processes. We distinguish two types of assembly features: handling and connection features. *Handling features* contain information on how to handle a component, and *connection features* contain assembly-specific information on a connection between components.

Handling features

Independently of where a component is positioned and oriented in the final product, some assembly-specific information can be stored for the component. For that purpose we use the handling feature; every generic component has such a handling feature. A *generic component* is a feature model of a component, independently of the actual place in the product. Within the handling feature, information on grippers that can be used to grasp the component is stored. For every specific gripper, different grip areas can be stored. Fixtures used to feed or fixture a component, and the number of components on a fixture, can also be specified in the handling feature.

Connection features

Feature models used to model single parts, show that some of the (design) features only exists for establishing a connection with a feature on another part, e.g. a pin on one part and a hole on the other. Usually, both parts are modelled independently, and by doing so no direct relation between the two features exists. It is better to simultaneously model the parts, in such a way that relations between features can be directly established. These relations between features can contain assembly-specific information.

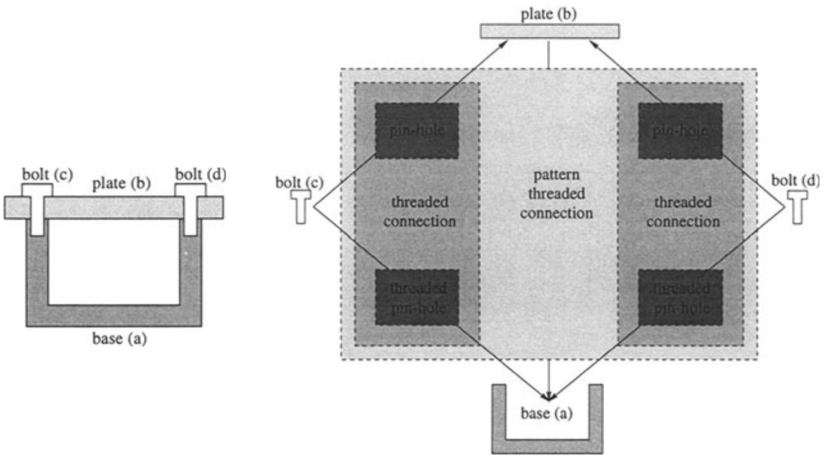
We offer the designer the opportunity to place specific connections, or connection features, between components. These connection features contain specific characteristics about the features on the components involved in the connection, and also additional assembly-specific information. The latter can, for example, be the insertion point, insertion path and final position of the connection. The insertion point gives the end position of the gross motion planning, the final position is the end position of the fine motion planning, and the insertion path is the trajectory between both positions. Also tolerances and characteristics on contact areas are known by the connection, and these can be used to find the internal freedom of motion, i.e. the set of motions that can separate the components.

In figure 3, an example is given of a product model with connection features. Dashed boxes represent connection features, and arrows give the components involved in the connection features. In the example, a compound connection feature is given; a *compound connection feature* can be subdivided into simpler connection features.

Some components in a product are specific in the sense that they are only needed to establish certain connections. For example, a bolt fastening two plates, is only present in the product because the plates must have some rigid connection. These specific components are called *agents*. Connection features contain information on which agents, if any, are involved in the connection. Removing the connection, will also result in removing the agents. In figure 3b there are two connections with agents: both threaded connections have a bolt as agent.

The connection features can also be used to represent *clustering* information within the product model. A cluster is a set of, possibly similar, components that share some assembly-specific characteristic, e.g. the same gripper or assembly direction (Boneschanscher and Heemskerk 1989). In a cluster, there is one of two opposite situations with respect to the assembly sequence in the cluster: either the assembly sequence is completely fixed, or it does not matter which sequence is taken. Take, for example, a large pin through a set of disks of increasing diameter, a so-called stack cluster. The disks do have a fixed assembly sequence, from large to small. On the other hand, a couple of bolts to fasten a plate, can also be seen as a cluster, a so-called sort cluster. Here no fixed assembly sequence to assemble the bolts is preferable. In figure 3b, an example of a cluster is given by the pattern threaded connection.

All information that is generic for a certain connection, can be stored within a connection feature. In this way, assembly-specific information about the connections is stored within the product model. The model can then be interrogated to retrieve this information.



(a) example product (b) connection features

Figure 3 Example of a feature model for assembly

4 USING FEATURES IN SEQUENCE PLANNING

The planning tools of section 2 can profitably use the information stored in the features of section 3. The basic idea of the planning tools can be the same, but the way assembly-specific information is retrieved is very different. Feature models can significantly decrease the complexity and the time used to retrieve this information. A consequence might be that the designer or engineer needs more modelling time, because he must add assembly-specific features to the model. However, this might be not that bad, because modelling will fit more to his way of thinking.

We will now show how features can be used in assembly sequence planning, using the concept of disassembly planning.

Each time we select a component as a candidate to be disassembled from the partial assembly. Using feature models, this is not done by trial and error, i.e. trying every component, but by using the connection features in the partial assembly, reducing the combinatorial explosion. Connection features can often give information on the precedences of the components involved in the connection, resulting in a priority for the selection of a component.

Connection features with agents, can quickly give the component to select first, because they 'know' something about the disassembly sequence. It is useless to try to disassemble a plate when it is still connected by a bolt, so the connection feature will present the bolt first, and after that the plate connected by the bolt.

A connection feature representing a cluster can directly give the predefined disassembly sequence for the components in the cluster (stack cluster), or can indicate that it does not matter in which order to disassemble the components (sort cluster).

As an example, the selection of the disassembled components in the product of figure 3a is given here. In figure 4a, the AND/OR graph is created without using features; some infeasible sequences investigated during the creation of the graph are also shown (these are marked with a cross). In figure 4b, the AND/OR graph is created using feature information (both agents and clusters), resulting in a much smaller graph.

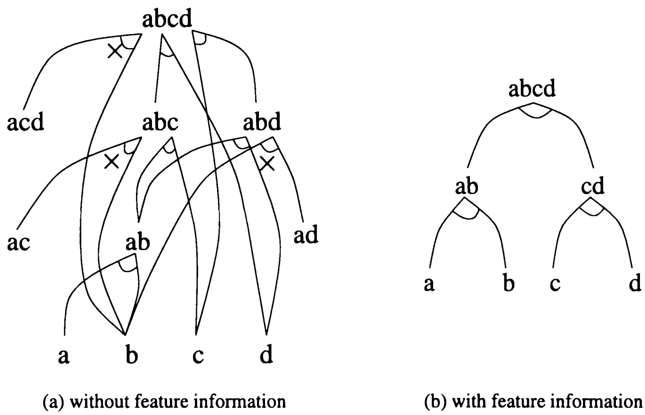


Figure 4 Creation of an AND/OR graph without and with feature information

Thus, by using agent and cluster feature information, the AND/OR graph can shrink significantly, resulting in a smaller number of precedences that must be checked. During precedence checking, other feature information can be used to further reduce computation time. This will be shown by some examples of the use of the planning tools described in section 2.

Finding other precedence relations

Connection features and handling features can directly generate contact areas on a generic component where a gripper cannot be positioned, because the areas are involved in contacts with the partial assembly, fixture or feeder. To find the possible grip areas on the remaining areas, the design features are used. Every design feature has a generic shape, so it can give the areas where a gripper finger can or cannot be positioned dependent on parameters of the finger. Gripper information can be found from the handling feature. However, using only design features for finding grip areas is not enough. Some low-level geometric computations are still needed, because design features can only give local grasping information, i.e. only information concerning the shape of the feature itself, independently of other features that can have influence. Although low-level geometric computations are still needed, using features results in better grip areas, and gives them faster, than using only these low-level computations. Details about this method are given by van Holland and Bronsvoort (1996).

In motion planning, especially in fine motion planning, features can be very useful. The connection features contain all information needed in fine motion planning: insertion position, final position, internal freedom of motion and knowledge about the best trajectory to assemble the component from insertion position to final position. Low-level geometric calculations are not needed anymore to find the type of connection and, for example, the insertion position.

An additional benefit in fine motion planning is that, for some connections, a fine motion path is directly available that could not even be found using geometric computations. Take, for example, a snap-fit connection. It is impossible to find an obstacle-free path from insertion position to final position, because the snap-fit connection depends on components that form obstacles, which are, however, not rigid, but deform during assembly. This assembly-specific information can easily be stored in a connection feature.

In stability analysis, the type of connection and the set of possible motions for a component are used. The connection feature can give information on the rigidness of the connection. If a connection is very loose, the internal freedom of motion can be checked for translational stability. If there is motion possible in the direction of the gravitational force, then the component is unstable. Details of this method are given by van Holland and Bronsvoort (1995).

All outcomes of the described tools using features, give precedence relations between the components in a product. These are used to prune the AND/OR graph as far as possible during its creation, resulting in a much smaller set of feasible sequences to investigate for the optimal candidate.

5 CONCLUSIONS

We presented a feature concept useful in assembly planning. Handling features contain information on how to handle a component, and connection features contain assembly-specific information on the connection between components. A designer or engineer can in this way store information about the assembly characteristics, which he already knows, in the product model.

This information can profitably be used during assembly sequence planning to reduce the number of assembly sequences to be checked. The information about agents and clustering, stored in connection features, can be used as a first step in the reduction. The next step is to use other assembly-specific information to determine precedence relations between components.

Some of the planning modules presented in this paper have already been implemented, giving successful results. On other modules, in particular the main sequence planning module, still some implementation work has to be done. Based on our experiences until now, we are convinced that this will show once more that features can profitably be used in assembly planning.

REFERENCES

Boneschanscher, N. and Heemskerk, C. J. M. (1989). Grouping parts to reduce the complexity of assembly sequence planning, *in* E. A. Puente and L. Nemes (eds), *Information Control Problems in Manufacturing Technology*, Pergamon Press, pp. 233–238.

Bronsvoort, W. F., Bidarra, R., Dohmen, M., van Holland, W. and de Kraker, K. J. (1996). Feature modelling for concurrent engineering, *Proceedings of the Symposium on Tools and Methods for Concurrent Engineering*, Technical University of Budapest, pp. 46–55.

Bronsvoort, W. F. and Jansen, F. W. (1993). Feature modelling and conversion - Key concepts to concurrent engineering, *Computers in Industry* **21**(1): 61–86.

De Fazio, T. L., Edsall, A. C., Gustavson, R. E., Hernandez, J., Hutchins, P. M., Leung, H.-W., Luby, S. C., Metzinger, R. W., Nevins, J. L., Tung, K. and Whitney, D. E. (1993). A prototype of feature-based design for assembly, *Journal of Mechanical Design* **115**(December): 723–734.

Gottschlich, S., Ramos, C. and Lyons, D. (1994). Assembly and task planning: a taxonomy, *IEEE Robotics and Automation Magazine* **1**(3): 4–12.

van Holland, W. and Bronsvoort, W. F. (1995). Assembly features and visibility maps, *in* A. A. Busnaina (ed.), *Proceedings of the 15th ASME International Computers in Engineering Conference*, Boston, pp. 691–697.

van Holland, W. and Bronsvoort, W. F. (1996). Extracting grip areas from feature information, *Proceedings of the 16th ASME International Computers in Engineering Conference*, Irvine, California. To be published.

Homem de Mello, L. S. and Sanderson, A. C. (1991). Representations for assembly sequences, *in* L. S. Homem de Mello and S. Lee (eds), *Computer-Aided Mechanical Assembly Planning*, Kluwer Academic Publishers, The Netherlands, pp. 129–162.

de Kraker, K. J., Dohmen, M. and Bronsvoort, W. F. (1996). Multiple-way feature conversion - opening a view, *(these proceedings)*, IFIP WG5.2 Workshop on Geometric Modeling, Airlie, Virginia.

Shah, J. J. and Rogers, M. T. (1993). Assembly modeling as an extension of feature-based design, *Research in Engineering Design* **5**(3 & 4): 218–237.

Waarts, J. J., Boneschanscher, N. and Bronsvoort, W. F. (1992). A semi-automatic assembly sequence planner, *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, Nice, France, pp. 2431–2438.

Wilson, P. R. and Pratt, M. J. (1988). A taxonomy of features for solid models, *in* M. J. Wozny, H. W. McLaughlin and J. L. Encarnaçao (eds), *Geometric Modeling for CAD Applications*, Elsevier Science Publishers B.V., Amsterdam, pp. 125–136. Proceedings IFIP WG5.2 Working Conference, Rensselaerville, NY, May 1986.