

A System Engineering Approach to Automated Assembly Planning

J.A. Barnett (JBarnett@NRTC.Northrop.Com)

Northrop Research and Technology Center
One Research Park
Palos Verdes Peninsula, CA 90274

ABSTRACT

The problem of generating assembly plans is examined from the viewpoint of totally automating the manufacturing process. In particular, it is assumed that automation is used to generate assembly plans from design data and robot programs from the assembly plans. The exploration is conducted by determining what knowledge must be shared between robot programmers and assembly planners, and by considering the impact, on assembly plans, of design variations within a simple product line.

PROCESS AND ASSEMBLY PLANS

A process plan specifies selections among manufacturing techniques and, thus, designates choices between fabrication methods such as extrusion and casting, and assembly methods such as screwing and riveting. An assembly plan is a sequence that incorporates the selected process steps and, to be complete, their detailed process parameters.

Process and assembly plans are intertwined because the knowledge to generate one is almost identical to the knowledge needed to generate the other: Process plans assume sequence and assembly plans respect these assumptions. The number of assumptions shared between the activities creating process and assembly plans is typically greater than between any other activities participating in a product's life cycle. For this reason, process and assembly plans are not distinguished below and, arbitrarily, the combined entity is called an *assembly plan*.

SUMMARY

In the typical enterprise, the macroactivities, design engineering, tool design, assembly planning, manufacturing implementation, and logistics are encountered in this order. Feedback is initiated only when errors are noticed in the output of a prior activity.

If an error can be corrected without feedback, it usually is.

The success of this pipeline scheme is enhanced if earlier activities know more about the requirements of those that occur later. Concurrent engineering [4] is one way to achieve this: practitioners from activities nearer the end of the chain interact with those performing functions nearer the beginning so that problems can be anticipated and eliminated before they occur.

An alternate approach is to integrate, in a single agent, more of the knowledge that is idiosyncratic to several activities. This approach is explored here with the assumptions that (1) a machine-readable design is available in some extended CAD representation, (2) our task is to design an automatic assembly planning system, (3) the output of this system will be consumed by other automated systems that generate tooling plans and robot programs, and (4) this is a small-batch environment. (Therefore, robot efficiency is not a primary consideration, though it must still be considered.)

The important conclusion is that assembly plans must incorporate some knowledge of tooling and robots and that design systems must anticipate the needs of assembly planners. These considerations are explored below by examining the nature of robot programs and determining the impact of varying design assumptions in an example product line.

ROBOT PROGRAMS

Assembly plans need to incorporate knowledge of robots if these machines are to be utilized effectively. Robot programs are examined in this section to find some clues about the nature of that knowledge. The task is difficult because languages used to program robots lack good constructs for abstraction. However, if we allow our imagination to make some generalizations, several patterns emerge.

The term, *robot*, is used to mean any tool that

can be programmed to exhibit a reasonable variety of behaviors. This class includes articulated arms, numerical control machines, and automated guided vehicles. The term could be usefully extended even further since much of the knowledge shared between assembly planners and robots programmers is still necessary when human touch labor is used instead of machines.

Program Constructs

In addition to declarations, the constructs used in robot programs can be roughly divided into three categories: motion commands, external control (I/O to agents in the work envelope, e.g., other robots and grippers), and numerical computations. The latter are the most prevalent while motion commands are the most complex because they are specified as a combination of goals, movement parameters (e.g., velocity), servo control parameters (e.g., joint interpolated motion), and guard conditions (e.g., not-to-exceed force thresholds).

The most striking thing about robot programs is the density of numerical constants that appear in them. In fact, the search to categorize the knowledge in robot programs boils down to the question: *Where do these numbers come from?*

Program Parameters

Parameters in robot programs are derived from knowledge of products, processes, and machines. An assumption, necessary to the success of automatic generation of assembly plans, is that only a summary of the latter is necessary. If fine-grained knowledge of machine-dependencies turns out to be needed, it will not be possible to reasonably separate machine programming and assembly planning tasks. Thus, the important discoveries are the origins of program parameters that are associated more with the product than the machine.

Product-specific geometry and topology¹ data is clearly needed by every robot program. Size, curvature, and connectedness specifications parameterize everything from motion commands to integrated vision systems [2]. It is assumed in many situations that this is all the information that is necessary.

Unfortunately, the assumption is false. What actually appears to happen is that other product-specific data gets incorporated in some early program version. Later, that program is reworked to do variations within the same product line and soon, the *raison d'être* for many of the program parameters is lost.

The knowledge sources obscured in the process are materials and functional requirements data. They imply, for example, tolerances and damage

¹ *Topology* deals with connectivity and relative proportions while *geometry* deals with absolute sizes.

thresholds which, in turn, determine the force constants that parameterize compliant motions. Materials and requirements knowledge, along with geometry and topology, is exactly what is needed to create combined process and assembly plans in addition to the robot programs that implement them.

DESIGN VARIATIONS

The most primitive assembly task is simple insertion. Tinkertoy,² a construction toy for children, features cylindrical wood sticks which are inserted into round holes, in multi-hole wood connectors, to grow larger assemblies. The large truss structures projected for use in NASA space stations [3] look like larger versions.

Next, the details involved in constructing assembly plans for Tinkertoy examples are considered. Then some assumptions about sizes and materials are modified to see what sort of impact the changes make. It becomes clear that simple design changes may cause assembly plan and robot program modifications even if process selection remains constant.

Figure 1 is a picture of a planar truss that is to be built from four short rods, R_1 , R_2 , R_3 , and R_4 , one long rod, R_5 , and four circular connectors, C_1 , C_2 , C_3 , and C_4 . The figure is the topological specification of a design that is used throughout the discussion in the remainder of this section.

Tinkertoys

An assembly plan generator for a product line consisting of things that can be built from Tinkertoy-like kits is entitled to enjoy certain assumptions. For the moment, assume that (1) components are made of wood, (2) rods are between fifteen and fifty centimeters long and between five and ten millimeters in diameter, and (3) connector diameters are between twenty and fifty millimeters. Later, these assumptions will be modified, to describe slightly different product classes, and the effects of the modifications will be examined. The current restrictions allow some size variations, but otherwise impose fairly tight design limits.

The specific task to be considered is generating an assembly plan for the design shown in the figure on the next page. Given the assumptions, the fabrication process selected is *purchase* and the assembly process is *worry*.³ These choices are likely to be correct whether the truss is assembled by human labor or a pair of robots—the assembly process selected demands either two hands or special jigs. The former is preferred when there are small-batch or product diversity requirements.

²Tinkertoy is a registered trademark of Play Skool Inc.

³*Worry* is an old-fashioned term for the type of twisting and pushing children use to assemble Tinkertoy models.

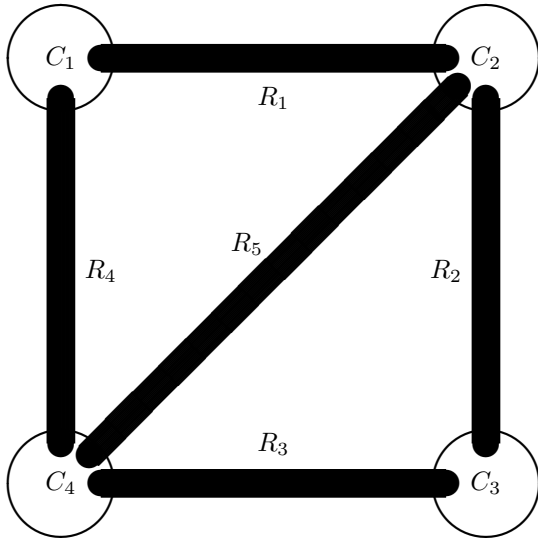


Figure 1: Rod and connector example.

The combined process/assembly plan is completed by specifying process parameters and producing a sequence of assembly steps. The parameters of component purchase—the selected fabrication process—are determined from the sizes, tolerances, and wood choices specified in the product’s design documentation.

The same design data is necessary to calculate the process parameters for the worry steps. The derived parameters are, insertion depth, twisting angle, and force data. If the worry process is used frequently, engineering handbook entries could be generated for it. Engineering handbooks archive relations between design parameters and process parameters so that experimentation and calculation, to determine the latter, need not be repeated for every application.

The assembly plan, in this simple case, is just an order in which to join components together by worry processes. Thus,

$$(1) \quad C_4 R_4 C_1 R_1 C_2 R_2 C_3 R_3 C_4 R_5 C_2$$

is an assembly plan. The interpretation is that C_4 and R_4 should be joined first, then that subassembly should be connected to C_1 next, etc.

This plan has the property that the hand (robot arm) bringing a new component to the assembly can continue holding it, after it is joined, while the other hand acquires the next component to be added. Thus, the plan is optimal in the sense that a certain class of useless motions are eliminated. The alternative plan,

$$(2) \quad C_4 R_5 C_2 R_1 C_1 R_4 C_4 R_3 C_3 R_2 C_2,$$

has the same optimality property as the first plan. In addition, R_5 , an interior component, is assembled before the perimeter is completed. Whether either optimization is important is a function of tool specifics, e.g., robot size and dexterity, and some design factors that are discussed below.

Material Changes

One way to modify the product class introduced above is to change the assumption that components are made of wood. Different materials can change the assembly order and, in some cases, even process selection. The effect of using plastic and metal are considered. The goal is not to increase knowledge about Tinkertoy variations. Rather, the point to be made is that assembly planners, ignorant about materials, are not particularly general.

Plastic components: If the requirement that components be wood is modified to require plastic but size restrictions remain constant, assembly plans are not likely to change much. However, there are at least two cases where plans need to be modified. The first case is small design tolerances. Plastic, unlike wood, binds and rods will not worry into tight holes. Thus, it may be necessary to augment the process specification to include shave-to-fit or some other customization operation.

The second case is where the coefficient of friction is not high enough to guarantee stability of the complete assembly and, thus, gluing or some other attachment process may be needed. Another factor to consider is that assembly steps should not put too much pressure on existing subassemblies. For example, plan 1, for the design shown in the figure, inserts rod R_5 *after* the perimeter is completed. This action might push the existing square outward and cause it to fall apart. Plan 2 is better because it does not have this problem.

Metal components: The change from wood to metal components can have more impact than the change to plastic. It is unlikely, though possible, that worry is still the process of choice. However, since much greater forces are involved, either special tools or jigs may be needed to do the insertions.

It is also possible that the best assembly process is to heat a connector, to expand its holes, then insert the rods. The optimal technique, for this strategy, is to insert all the rods in one connector, then do another connector, etc. Thus, a reasonable assembly plan for the design shown in the figure is

$$(C_4 R_4 R_5 R_3)(C_2 R_5 R_1 R_2)(C_1 R_1 R_4)(C_3 R_2 R_3),$$

where parenthesized groups represent heating and insertion cliques. When a jig is used, only one robot may be needed instead of two. If so, motion op-

timizations, like those reflected in plans 1 and 2, become moot.

Size Changes

The results of varying the size assumptions are analyzed in this section to show that the contents of assembly plans are a function of component sizes even if topology and material are constant.

Larger trusses: Assume that the rods in the example have lengths of several meters and diameters of several centimeters. The connectors are correspondingly larger. We can no longer take worry to be the *sine qua non* of processes because the larger twisting forces, necessary to overcome friction, may break the rods. Therefore, (1) designs may need to be modified to reduce (increase) rod (hole) diameters and (2) assembly plans may need to be correspondingly modified to include combinations of simple non-twisting insertions and attachment processes such as gluing. Thus, the process specification portion of the plan may change.

The assembly sequence can change too. If the dimensions are large enough, manufacturing may need to use jigs either because (1) the components are too cumbersome or (2) free motion of the rods may cause them to snap when their length/diameter ratio exceeds a threshold.

Jig design and limitations can virtually determine assembly sequences. For the example in the figure, jig use might dictate that the diagonal rod assembly (R_5, C_4, C_2) be clamped before attaching a perimeter triangle. Under this assumption, one reasonable assembly plan is

$$([R_5 C_4 C_2] R_4 C_1 R_1 C_2)^{-1} R_3 C_3 R_2 C_2,$$

where the inner group in square brackets is a jiggling operation, and the inversion exponent means that the upper triangular subassembly, $C_1 C_2 C_4$, should be flipped so that the lower triangle, $C_2 C_3 C_4$, can be built using the same jig facilities.

Smaller trusses: Shrinking can have as much impact on assembly plans as expansion. If either rod diameter or both rod length and rod diameter are reduced from Tinkertoy size by an order of magnitude, there will be little structural integrity.

The choice of processes and assembly sequences must account for this lack. Some combination of micromanipulators and jigs will be needed. Jigs can introduce the same kind of variations seen above when sizes were increased.

Other sorts of considerations need to be accounted when the ratio of the size of the tool to the size of the parts change. In the case considered here, it is probably a good idea to avoid working on the interior, after the exterior is completed, because the tool may not fit in the restricted space.

Thus, plan 1 is probably flawed while the others are probably acceptable.

CONCLUSION

Automatic assembly planning systems share a bit of ignominy with the products of AI laboratories—both are reputed to work only in toy worlds. Though true, the criticism is partially unfair because parameter suppression is a legitimate experimental technique to reduce problem complexity.

The toy world syndrome occurs when large numbers of constraining assumptions, imposed to make prototype development tractable, become deeply wired into systems and the dependencies that they engender are not fully appreciated. Complexity is reduced but the knowledge used to do so is lost [1]. The early work that was so impressively demonstrated does not generalize and no one is sure why. The reasons for this problem and ways to avoid it were sought above.

The analyses, using the extended Tinkertoy example, show that all enterprise activities must share product assumptions and that each activity must know a fair amount about how the others do their work. It is hard to imagine a product class more primitive than this one which relies only on a simple insertion operation. Yet, even in this case, the amount of knowledge that must be shared among activities is substantial.

The above analyses also demonstrate that topology is only a small part of the data that drives assembly planning and robot programming. Knowledge about materials and absolute sizes has just as much impact on process selection, assembly sequences, and tool particulars. If automatic assembly planners are to become part of integrated enterprise systems to assist in the production of more complex items, the range of design parameters considered must be increased.

References

- [1] Barnett, J.A., An Aler's lament, *AI Magazine* 7(2) (1985).
- [2] Hoffman, R., Keshavan, H.R., and Towfiq, F., CAD-driven machine vision, *IEEE Transactions on Systems, Man, and Cybernetics* 19(6) (1989) 1477–1488.
- [3] de Mello, L.S.H., Desai, R.S., Assembly Planning for large truss structures in space, *These Proceedings*.
- [4] Wood, R.T.(ed.), *Proc. Second National Symposium on Concurrent Engineering*, Morgantown, WV (Feb. 7–9, 1990).