

Intelligent Decision Support for Assembly System Design

*J. A. Hernandez, T. J. Peters, D. E. Whitney, S. C. Luby,
R. E. Gustavson, H. W. Leung, P. M. Hutchins, T. L. De Fazio,
J. L. Nevins, A. C. Edsall, R. W. Metzinger, and K. K. Tung*

The design of an assembly system is a complex problem, often requiring large teams of engineers with significant experience and expertise. Based on studies of the product design, these teams must select an assembly sequence, generate a corresponding process plan, identify the resources capable of completing the tasks in the process plan, and group these resources into a system of workstations to accomplish the required assembly operations. The resulting system must be economically viable but have sufficient capacity to meet production requirements.

The successful, timely design of such an assembly system depends on the team's ability to effectively share and communicate large volumes of information and have rapid feedback concerning cost, producibility, quality, and other important engineering criteria. Unfortunately, few available tools focus on fabrication or assembly: Those tools that do are often unable to share data. As a result, design teams are forced to do much of the work manually and are unable to effectively share information. Even large manufacturers find their teams spending considerable efforts just to enter the same data into multiple computer systems.

Much work has been done to address various aspects of the assembly

system design problem. For example, systems have been developed for modeling (Dixon 1988; Pratt and Wilson 1985; Luby, Dixon, and Simmons 1986), assembly design (Boothroyd and Dewhurst 1987), sequence selection (Lui 1988; Homem de Mello 1989), process planning (Delchambre, Coupez, and Gaspart 1989), and assembly line design (Gustavson 1988; Graves and Holmes 1988; Coopriider 1989). As a result, much has been learned about these particular areas. However, little work has been done to integrate these systems. Work on integrating these systems has required limiting assumptions, such as the use of embedded domain-specific knowledge (Sriram et al. 1989), or application to parametric products only (Phillips and Aase 1990).

The goal of this work is to provide assembly system design teams with an integrated environment that is capable of intelligent decision support for work on complex assemblies. The system was developed through the combined efforts of a group of software developers and a team of electric, industrial, and mechanical engineers. The engineers have worked on design and assembly for 20 years, doing both basic research and consulting work with several large manufacturers. The choice of functions for the system was driven by the engineers' need for a set of tools to handle recurring problems in their consulting work. The tools address issues that occur over a wide range of industries and that are critical to the design of technically sound, yet economical assembly systems.

Cognitive Model

In addition to the basic functions, we felt it important for the system to incorporate basic principles of cognitive modeling. We identified five principles that have a strong impact on the flexibility and usability of the system. According to these principles, the system should perform five functions: First, the system should use the designer's customary vocabulary, thus raising discourse to the conceptual level and making design more natural. The result is an increase in user productivity and a decrease in the probability of errors because complex calculations are automatically done by the computer. Second, the system should provide incentives for the user, for example, permitting the creation of designs that will stand scrutiny on design for manufacturing and assembly and cost criteria. This approach will assist users in learning appropriate methodologies and motivate them by providing rapid, positive feedback. Third, the system should maintain a record of the user's intent in making design decisions. This information can be used at a later time to understand the varied and complicated interactions among different aspects of the design. In addition, we are investigating knowledge

representation schemes that can make the designer's intent accessible to application modules for analysis and recommendations. Fourth, the system should allow the user a choice of techniques and methodologies to follow. This function lets the user explore various approaches to a problem, which could lead to alternate problem formulations and better solutions. Fifth, the system should support modules that are themselves stand-alone systems, providing users with flexibility on how to use the system.

These five principles, together with the functional requirements, led us to the development of the architecture shown in figure 1. The system consists of six major modules: a feature-based modeler, a constraint-identification module, an assembly sequence selection module, a process planner, an assembly system synthesis module, and an economic analysis module.

The *feature-based modeler* allows the user to construct a product design that includes assembly information that indicates the mating relations between the parts. The *constraint-identification module* then uses this information, together with the results of a guided question-answer session with the designer, to determine constraints on the assembly process. Once all the constraints are known, the designer can invoke the *assembly sequence generator*, which generates the search space of all the feasible assembly sequences. The designer can then prune the sequences through the use of various automated techniques or manual editing. An *assembly process plan* can then be generated for each selected sequence. Each task in the plan includes information on part and sub-assembly orientation, size, weight, and relative task difficulty. This information can be used by the *assembly system design module* (Gustavson 1988) to select assembly resources and create a least cost system to meet required production rates and investment targets.

The user can begin with any module or can bypass the use of any module and let the system act as an expert to generate solutions to the modules being bypassed. Moreover, the user can investigate alternatives by studying the effects of making revisions to the original design, selecting a different assembly sequence, or changing cost and production requirements.

In the rest of this chapter, we discuss each of these modules using a moderately complex design to demonstrate the capabilities of the system. Then, we place the work in perspective and discuss related research.

The system combines a variety of technologies; is implemented in I-DEAS, Lisp, and C; and runs on Sun 3 workstations. I-DEAS provides a strong computer-aided design (CAD) environment over which we were able to build a feature-based modeler. C is used for modules requiring many computations but no symbolic processing. The major portion of

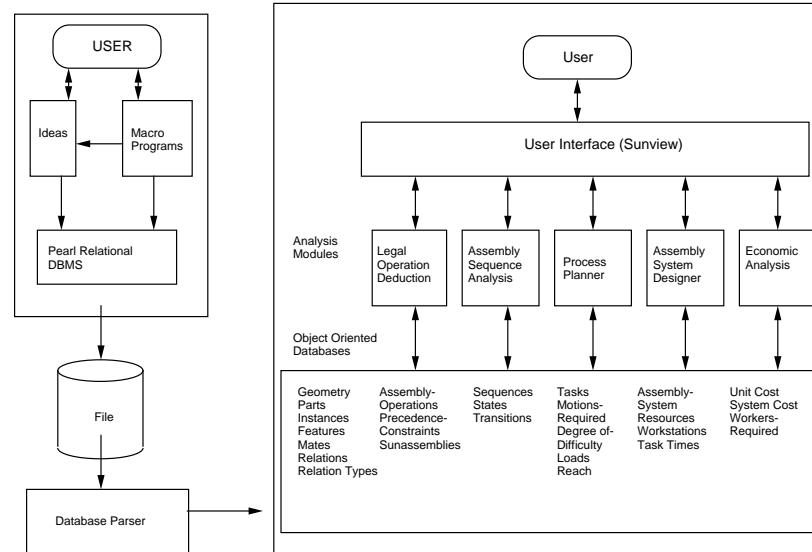


Figure 1. System Architecture.

the system is written in Lisp to support an object-oriented database representation of both product and economic information and support symbolic reasoning. We chose to use only one platform to avoid networking issues.

Building the Product Model

Our approach to product modeling begins with *feature-based design*, a technique that permits designers to express design intent while they create the product geometry (Dixon 1988; Pratt and Wilson 1985). This approach both requires and permits the designer to think beyond mere shape and explicitly state what portions of a part are important and why. In addition, feature-based design brings modeling closer to the user's conceptual level, making design more natural. It also results in a format that is well suited to object-oriented technologies, making it simple for the program to reason about the design to generate and evaluate assembly plans.

Feature-Based Design

We developed a feature-level parts and assembly modeler on top of the I-DEAS package using SDRC, Inc.'s, IDEAL programming language and PEARL relational database facility. This feature-based modeling layer al-

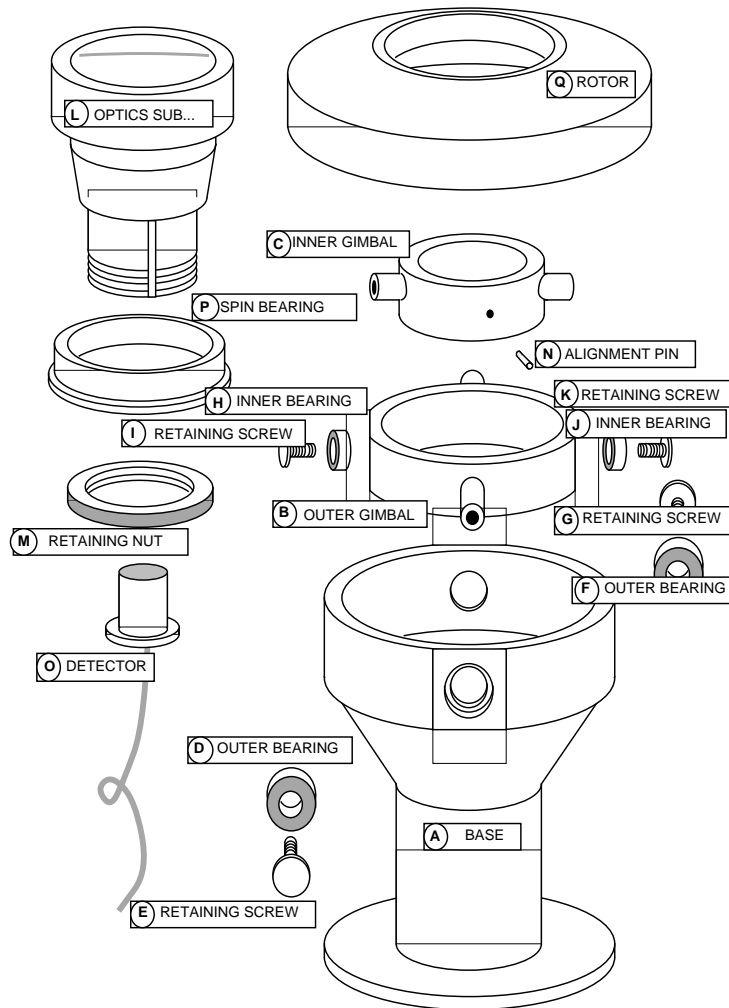


Figure 2. Exploded Parts View for the Seeker Head.

allows a user to create geometry using higher-level commands and builds a database of feature-level information as the user works on a design. The downstream analysis applications (assembly process analysis and economic evaluations) can then access this database and reason about the designer's intent.

The feature-level modeler is divided into two main modes: part modeling and assembly modeling. In *part modeling*, the designer can create product components as generic pieces of geometry that are augment-

ed and modified by subsequent feature modeling. Each component can have multiple instances. For example, a $\frac{3}{8}$ -inch bearing part can have a left and right instance with unique names and positioning. The designer can also build components out of features that are represented by existing geometry. This capability handles cases where it is simpler to create some geometry using primitive solids (for example, by revolving a profile) or allow for feature types (for example, Flats) not easily created from primitive solids. Once the basic part data are entered, the modeler computes further information required for the application modules, for example, assembly weight, assembly center of gravity, assembly bounding box, and feature reference directions.

Figure 2 shows an exploded parts view of an 17-part product we used as a test case. It is the mechanical structure of a generic seeker head, a complex component of tactical air-to-air missiles.

After creating the parts, the user switches to the *assembly modeler* and defines the assembly as a set of mated part instances. In this way, the assembly topology is explicitly described by simply specifying the mates between component part instances. The system uses the database information about reference directions and locations of the features to align parts in their proper relative location and orientation for the final assembly state of the product. Figure 3 shows a cross-sectional view of the final assembled product.

The designer is also able to provide information about interface features. For example, is a mate composed of a spline in a through hole or a peg in a blind hole? Typical assembly information provided by the user includes feature-mate type, presence of lubricants or adhesives, and permission to grip on the feature or attach a fixture on the feature.

During modeling, the program layer builds a database of the higher-level part and assembly information using the I-DEAS PEARL relational database facility. After modeling, this database is downloaded to the rest of the system and converted into an object database. The system then uses this information to find clearances and degrees of freedom for mates, relative extraction directions for mating features, and total weight and bounding-box size for the product and all subassemblies and distances between features on different parts.

Assembly Information

Based on the information contained in the database downloaded from the feature-based modeler, the system generates a graph of all mates that connect parts in the assembly. Because mating features are linked by mating relations, and features have links to corresponding assembly parts, the system can traverse these links to identify all mates between

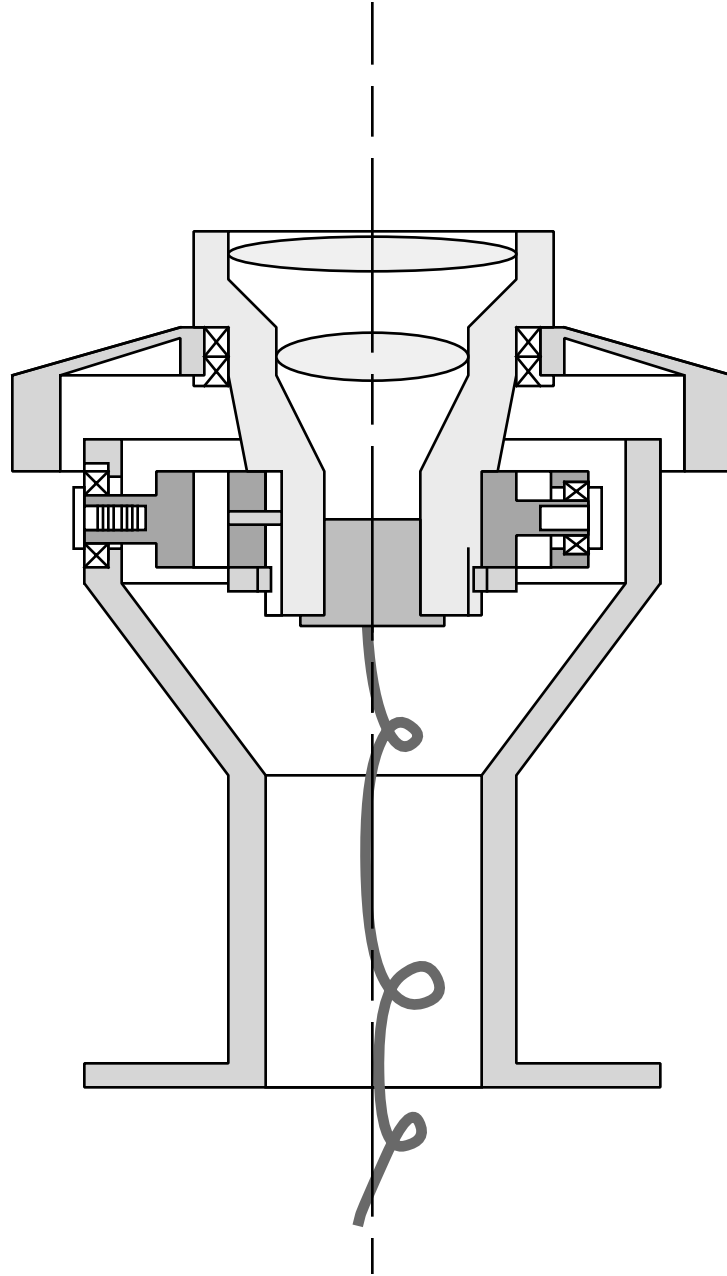


Figure 3. Cross-Sectional View of the Assembled Seeker Head.

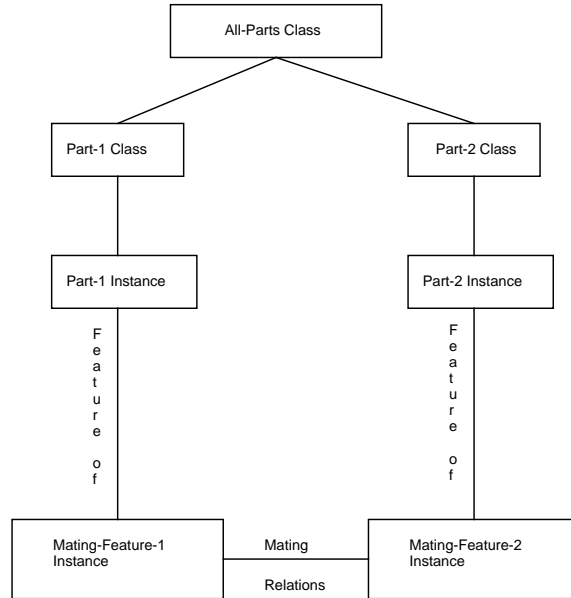


Figure 4. Representation of Relations between Components.

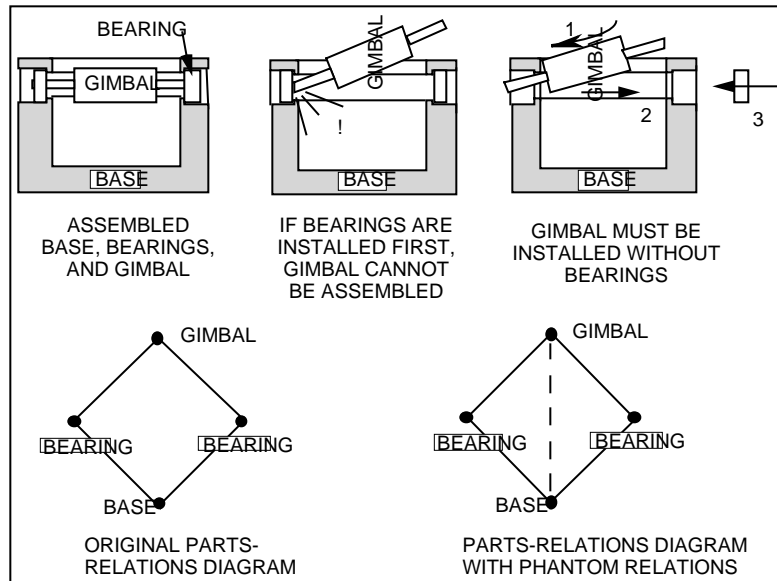


Figure 5. Assembly of Parts That Do Not Directly Mate.

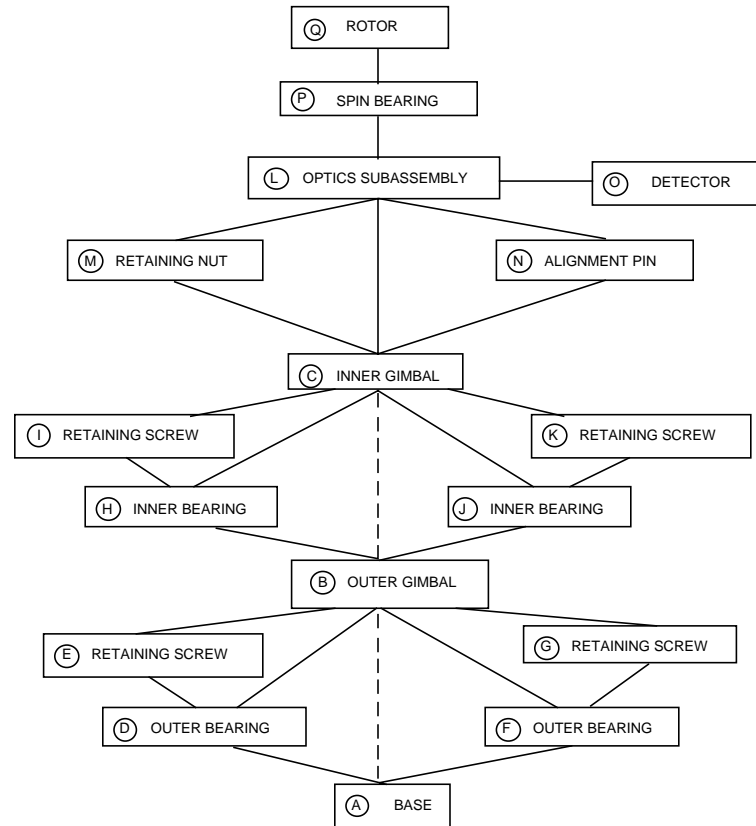


Figure 6. Parts Relation Graph for the Seeker Head.

components. Figure 4 demonstrates the representation of mates between assembly parts as relations on the corresponding part features.

In addition, it can happen that the relation between parts that do not mate with each other must also be added to the graph. This situation occurs when temporary fixtures are necessary to align one of the parts with an existing subassembly that contains a second part. This approach is sometimes necessary for full product assembly to be possible. Figure 5 demonstrates this situation for the gimbal and base components of the seeker head.

The graph showing all mating and nonmating relations between seeker head assembly components is shown in figure 6. Note that the graph consists of 24 mating relations and 2 nonmating relations. There is 1 nonmating relation between the inner gimbal and the outer gimbal and 1 between the outer gimbal and the base.

Assembly Sequence Selection

The choice of assembly sequence is critical to the proper design of an assembly system. The system provides the user with a choice of modes to use for sequence selection: The *assistant mode* gives the user full flexibility in exploring alternate sequences, and the *expert mode* uses heuristics to automatically select a sequence for the user. Alternatively, the user can directly enter a sequence.

Representation

Figure 7 shows a graphic display of some partial feasible assembly sequences for the seeker head. Each node in the graph represents an assembly state. The node in row 0 (starting at the top of the figure) represents the null assembly, and the node in the last row (not shown, but this terminal node would have all its cells blackened) represents the completed assembly. Consecutive assembly states are represented as nodes linked by accomplished relations. Each path from row 0 to the bottom row identifies a unique assembly sequence. Figure 7 depicts only a small percentage of all the assembly sequences. The full diagram for the seeker header contained 15 rows, the widest row contained 34 columns, and the total number of possible sequences exceeded 20,000.

Assembly Constraints

Three major constraint categories were identified for assemblies (Homem de Mello 1988). *Geometric-feasibility* constraints require collision-free paths joining subassemblies. *Mechanical-feasibility* constraints require the establishment of attachments acting on contacts of the subassembly decompositions. Finally, *stability constraints* require that subassemblies maintain their relative position and do not spontaneously break contact.

Many of the geometric and mechanical constraints can be determined from the relations among parts and the mating directions, but the automatic generation of all the constraints inevitably requires a general solution to three-dimensional path planning.

Because the designer can usually solve such problems almost instantly if shown a sufficiently realistic drawing of the product parts, we chose to involve the user to obtain assembly constraint information that cannot logically be deduced from the features (Baldwin et al. 1991). This task is accomplished by guiding the user through an illustrated question-answer dialogue. The number of questions asked of the designer varies considerably from product to product; for the seeker, 58 questions were asked. The computer answered 20,3754 other questions itself using feature information.

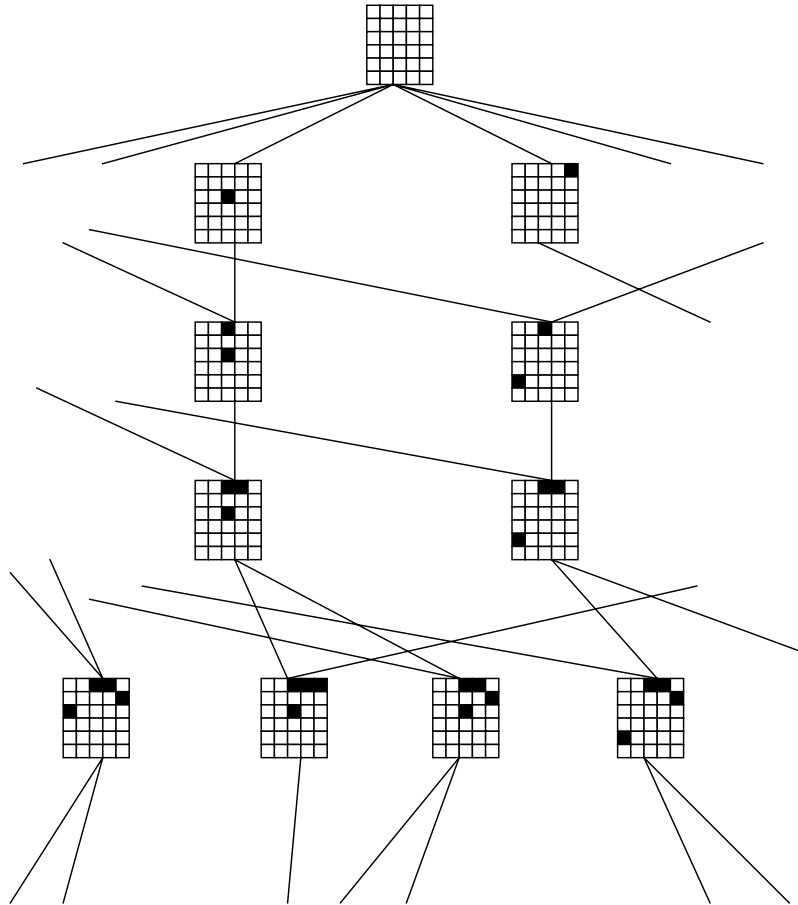


Figure 7. Some Partial Assembly Sequences for the Seeker.

Assistant Mode

The assistant mode first generates the search space of all physically possible assembly sequences (Lui 1988). This set consists of all sequences meeting the geometric and mechanical feasibility constraints. The search space can contain tens of thousands of sequences even for products of only moderate complexity and part count. Therefore, it is crucial to provide the user with utilities to navigate through this search space and tools for pruning it to a manageable size. The user can become acquainted with the possible assembly sequences by scanning and zooming options for the display. In addition, by selecting a state on the graph, the user can see a picture of the assembly represented by

the state. Similarly, the user can select a pair of connected states to see a picture of the corresponding assembly move.

Once familiar with the search space, the user can edit it. Manual editing facilities allow the user to delete states, transitions, or entire sequences from the state space. The user can also prune the space by specifying logical constraints on the system, for example, constraints that require that a specified group of mates not be established simultaneously or require an ordering on mates.

Automated pruning facilities permit pruning at a coarser scale. Possibilities include the automatic elimination of sequences representing equivalent assembly line topologies (that is, have the same part groupings) (Amblard 1989).

A program (Abell 1989) also addresses the stability constraint problem and finds the sequences requiring the fewest fixtures and reorientations. This capability lets the designer evaluate the inherent cost of fixture and reorientation operations required by the given design. By using these tools, the designer can thoroughly consider the possibilities and arrive at a reasonable set of sequences in less than an hour for products similar to the seeker head in complexity and part count.

Expert Mode

As part of a related research effort, one of the authors also implemented an expert system for sequence selection. The process consists of three main steps: First, a base component is chosen. Second, an exploded view of the assembly is established. Finally, a sequence is incrementally generated.

To select the base component, the system goes beyond simple mate count (Ko and Lee 1987) and reasons about the trade-off between size, weight, number of mates, and difficulty of establishing these mates.

The exploded view of the assembly is then generated using the information obtained from the feature-based modeler. This step generalizes the approach used for uniaxial products (Kroll, Lenz, and Wolberg 1989a, 1989b) and requires determining each component's assembly direction with respect to the base component. The component's relations (that is, mates), final assembly location, geometric center, and bounding box are used to make this determination. The result is an ordering of all components along the six possible three-dimensional directions.

From the exploded view, the assembly sequence can be determined. The program uses the heuristic that assembly should proceed along only one direction at a time whenever possible. This approach avoids reorientations. In addition, the sequence must meet the geometric, mechanical, and stability constraints previously discussed.

1	Attach SPIN_BEARING-P to pallet.	.
	Type = Place	DoD = 2
	Mot Req = Z	Load = 0.2281
2	Insert ROTOR-Q	
	Type = Assemble	DoD = 3
	Mot Req = Z	Load = 1.7700
3	Rotate pallet/trunion.	
	Type = Automatic-reorient	DoD = 1
	Mot Req = X, Y, Z	Load = 4.3297
4	
	
:		
:		
:		
:		
36		

Figure 8. Partial Process Plan for the Seeker.

Assembly Process Planner

The assembly process planner builds a complete assembly plan for the sequences previously selected. Additional steps added by the process planner include orientation, attachment of fixtures, application of adhesives, oven curing of adhesives, and torquing of bolts. Moreover, each step in the process plan contains load, reach, and motion requirements. Heuristics determine task difficulty based on type of mate, sizes and weights of parts, and number of features that mate simultaneously.

The data necessary to add this information can be determined from the geometric features and information supplied by the designer. Figure 8 shows a plan for the seeker head. When the software finishes generating the process plan, the designer can review it and make any changes desired.

Assembly System Designer

The last step accomplished by the system is to convert the assembly process plan into a list of equipment and tools together with the plan steps that each piece of equipment will accomplish. The resulting plan should result in the lowest unit cost of assembly.

Candidate Equipment Selection

The system selects candidate equipment capable of completing the steps in the process plan. Possible equipment includes manual assembly, maintenance, robots, and conventional fixed-assembly automation. The selection process is carried out heuristically, using multiple sources of information as constraints on equipment costs and speeds. For example, bounding-box size of subassemblies, weight of parts, and type of mate are used to identify constraints on the type of equipment required. Larger, heavier parts might require longer reach or more stability and might use more costly equipment, more assembly time, or both. The estimated task difficulty is also used to rule out certain types of equipment and determine equipment cost. Finally, economic and production data, such as cost of labor, desired annual production quantity, and information on investment return requirements, are used as filters on equipment cost and speed. See Allan et al. (1990) for more information. The system also reasons about the trade-offs in cost, time, and flexibility arising from the assignment of several tasks to flexible equipment such as manual labor or robots that can do more than one task.

Feasible Workstations

A *workstation* is defined as a resource that can perform consecutive tasks. Feasible workstations can perform all their tasks within the maximum time allowable by system demand. The system finds all feasible workstations by taking into account the yearly production demand, the cycle time required for the workstation to complete each of its tasks, the time required to change tools, the time incurred in moving partial assemblies along workstations, and the amount of extra time the workstation needs for maintenance and repairs.

Assembly System Synthesis

The optimal solution can be calculated by finding the least cost combination of feasible workstations that can complete the product assembly.¹ The system searches the space of solutions incrementally by building the solution table for the system, beginning with the solution table for the null process plan. Solution tables (ST) are built from the workstations (WT) using the following relationships:

$$ST_0 = \phi$$

$$ST_i = \text{Merge}_{j=0}^{j-1} [ST_j + WT_{j+1, i-j}]$$

In these equations, the first index specifies the initial task of each workstation, and the second index specifies the number of tasks accomplished by the workstation. The merge function takes the lower envelope of the graph of the cost curves.

If the resulting solution table for the entire process plan is null, then no assembly system design can complete the process plan. This incomplete action can occur when there is a particular task that cannot be accomplished quickly enough by any resource to meet demand.

To find the optimal system, the solution table is scanned for the cycle time that minimizes the total system cost. This time corresponds to the workstation operation time that minimizes total system cost. For each workstation, the resulting plan identifies the resource used, the tasks performed, and the costs incurred. In addition, the user is provided with system characteristics, including production rate and capacity, number of shifts, unit cost, and system cost. Figure 9 shows the system design for the seeker.

This system is guaranteed to be least cost for single-resource workstations. However, the overall least cost system can include workstations with more than one resource. To find this system, the least cost single-resource workstation system is used as a benchmark. Then, each workstation cost table is repeatedly modified to reflect the addition of an extra copy of the original resource. When the total cost for each workstation exceeds the original benchmark, the augmented tables are combined. The resulting system is guaranteed to have the overall least cost. The previous procedure can be expensive computationally. Therefore, we have been considering modifying it to test the difference in cost for the workstations after each iteration. The process can then be terminated as soon as the difference gets small enough to be insignificant instead of waiting until the new workstation cost exceeds the benchmark. This technique would yield a near least cost system

Station	Begin-task	End-task	Resource	H-cost	T-cost
1	1	4	Prog	45000	98941
2	5	9	Fixed	0	77424
3	10	14	Prog	50000	115914
4	15	19	Prog	45000	104609
5	20	24	Prog	50000	113645
6	25	29	Manual	200	233892
7	30	34	Manual	200	233183
8	35	35	Fixed	0	60176
9	36	36	Manual	200	144805

Number of shifts	1.00	shifts
Unit costs	11.83	\$
System cost	384546.54	\$
Production rate	48.11	units/hr
Production capacity	277119.85	units/yr
Utilization	8.36	%
Number of stations	9.00	stations

Figure 9. Partial Final Assembly System for the Seeker.

that would be within acceptable limits of accuracy. Moreover, this limit could dynamically be set by the system user.

System Evaluation

The system described in this chapter provides users with an intelligent decision support environment linking feature-based modeling, computer-aided process planning, and assembly system design. The system required an effort of approximately eight person-years—four to develop the specifications for meeting the needs of the engineering team and four to implement and test.

The modules from the system have been used in consulting work on major industrial products both separately and in various combinations. We find that using the system results in increased productivity. Moreover, the ability to work on separate modules using a centralized database provides consistency of information and results in significant time savings.

Therefore, the engineering team is able to reduce the time required to evaluate product designs and recommend factory designs.

Our feature-based interface allows a complex sequence of solid modeling Boolean commands to be replaced by a single feature creation command, such as “insert chamfered hole.” Subjective evidence with this simplified interface supports time savings in the model creation process. Furthermore, when features are used as a knowledge representation, considerable time savings can be achieved in downstream processes.

For example, in the seeker head example cited in the subsection on assembly constraints, the features were an essential knowledge representation for the automatic generation of geometric and mechanical constraints. The presence of these constraints allowed a reduction in the question count requiring user response from 203,754 to 58. Assuming that a typical question requires 15 seconds for a human response, the corresponding time reduction is from approximately 21 person-weeks to 14 person-minutes.

Our only analysis of the seeker head used our intelligent decision support system. We do not have time estimates regarding the seeker head without such assistance. However, such benchmark information is provided by an example assembly (De Fazio and Whitney 1987); this assembly is named *assembly from industry* (AFI). Furthermore, its complexity is roughly comparable to that of the seeker head, so we contend that it provides for a consistent comparison. A user of our system was able to select a preferred AFI assembly sequence within an hour. Previously, the required time was approximately eight person-weeks.

The cumulative efficiency gains can be translated into substantial direct monetary savings. Totaling the time savings as approximately 30 person-weeks and assuming that the analysis would be performed by a specialized consultant at a cost of \$1,000 each day, a direct savings of \$150,000 could be realized. Moreover, there are hidden yet significant savings that are difficult to directly represent in terms of dollars.

The richly expressive feature-based interface is much more natural for mechanical designers than the formal Boolean operators of solid modeling. This approach not only gives the designer greater confidence in a complex model but also permits verification of the functional aspects of the design.

The factor of 21 person-weeks of human interaction is so obviously prohibitive that in practice, all such constraint questions are not answered in the absence of our intelligent decision support system. However, the attendant cost is that the assembly system design is based less on comprehensive analysis and more on the designer's personal heuristics (often unarticulated), running the risk that a preferred assembly system design can be overlooked because of the complexity of the analy-

sis. Thus, our system permits comprehensive analysis on assembly systems of greater complexity than might otherwise be attempted.

Each of the time savings cited can have a ripple effect through the product process cycle. All savings were achieved within the conceptual phase of the assembly system design. Hence, the entire product manufacturing cycle can begin earlier. Furthermore, the rigor of the analysis argues that the assembly system design is "right the first time." The attendant savings by avoiding redesign and rework can easily total months or even years. Recently, Boeing was able to utilize a digital preassembly method to halve the number of design changes required during production (Stix 1991). This preassembly software was far less sophisticated than our intelligent support systems. In fact, the Boeing software included only vendor available CAD tools, with no intelligent aids. Our more sophisticated system should provide even greater dividends.

In fact, the considerations of the subsection on assembly system synthesis will permit an optimal assembly system. Many existing assembly systems perform reasonably well but are not optimal. Their deviations from optimality can be so subtle that they are not even noticed by trained observers. However, this suboptimality reflects a real cost that is compounded over the life cycle of the assembly system. In summary, this intelligent decision support system can provide a significant competitive advantage in an arena where time to market is often the most critical variable in the success of a product.

An additional benefit is that the engineers report a greater level of confidence in their findings. By taking advantage of rapid feedback from the system, the engineers are able to explore alternate assembly opportunities, discover assembly problems inherent in the design, and correct them at an early stage. Finally, because only the cost models need to be updated to reflect industry-specific variations, the system is easily maintainable by industrial engineers with little if any assistance from knowledge engineers.

Future Work

An interesting opportunity is to enhance the functions of the system with other, existing technologies. Important work has been done on assembly design using both quantitative and qualitative techniques (Laszcz 1984; Poli and Fenoglio 1987; Boothroyd and Dewhurst 1983). These techniques could be added as a critic component that suggests improvements to the design created through the modeler. Because these techniques can lead to conflicting recommendations, the system can evaluate the suggestions using final system cost as the determining factor in deciding among conflicts. By using an intelligent control

mechanism, the system could then iterate through this process, resolving conflicts among sets of recommended design changes to find the optimal set of improvements to the product design.

A research area of particular interest is the development of algorithms for deciding among alternate testing strategies (Pappu 1989). The choice of testing strategies has a significant impact on the design of an assembly system. In particular, in-process testing of partially completed assemblies can help detect problems sooner and more easily. This approach leads to earlier, simpler, less expensive repairs. However, choosing an appropriate strategy is a complex problem. It requires reasoning about trade-offs among the likelihood of failures, the cost of test equipment, the cost and time of repairs, and the interaction of test capabilities with possible failures. Further complications arise because some tests are not able to determine the causes of failures. Also, certain failures can only be detected after full assembly. Further empirical and theoretical work is needed to develop algorithms for making the correct choice.

An objective method to quantify total time saved would be to develop two assembly system designs—one using our system and the other not. Our fiscal constraints argued strongly against such an experiment. Thus, our measures of time savings were based on less comprehensive data. Although we believe that the examples cited are indicative of substantial time savings, it would still be instructive to conduct such a comprehensive experiment.

In summary, the system provides a robust, flexible, integrated environment for the design of assembly systems. The system has had the benefit of being useful for practical work. We believe the system provides users with a unique environment for making sophisticated decisions across many levels of the product life cycle. In the future, we hope to continue to increase the capabilities of the system.

Acknowledgments

The authors would like to thank C. S. Draper Laboratory for supporting the project and permitting those authors who have left the laboratory since this work was done to write this chapter. The authors also gratefully acknowledge the generosity of the researchers of the Alpha_1 Modeling System, University of Utah, Salt Lake City, Utah, for providing the chapter cover illustration depicting mating features in an assembly.

Note

1. See Coopriider (1989) for a good discussion of generic cost functions and Allan et al. (1990) for a more specialized study in the space industry.

References

- Abell, T. E. 1989. An Interactive Tool for Editing and Evaluating Mechanical Assembly Sequences Based on Fixturing and Orientation Requirements," Master's thesis, Mechanical Engineering Dept., Massachusetts Institute of Technology.
- Allan, D. C.; Denktsis, G. F.; Eppinger, S. D.; and Jakiela, M. J. 1990. An Engineering Model for Predicting Manufacturing Costs of Aerospace Components. Presented at the ASME Winter Annual Meeting, Dallas, Tex., 25-30 November.
- Amblard, A. G. P. 1989. Rationale for the Use of Subassemblies in Production Systems. Master's thesis, Operations Research Program, Massachusetts Institute of Technology.
- Baldwin, D. F.; Abell, T. E.; Lui, M-C M.; De Fazio, T. L.; and Whitney, D. E. 1991. An Integrated Computer Aid for Generating and Evaluating Assembly Sequences for Mechanical Parts. *IEEE Transactions on Robotics and Automation* 7(2): 78-94.
- Boothroyd, G., and Dewhurst, P. 1987. *Product Design for Assembly*. Wakefield R.I.: Boothroyd and Dewhurst, Inc.
- Boothroyd, G., and Dewhurst, P. 1983. Design for Assembly: A Designer's Handbook, Dept. of Mechanical Engineering, Univ. of Massachusetts.
- Coopriider, C. B. 1989. Equipment Selection and Assembly System Design under Multiple Cost Scenarios, Master's thesis, Sloan School of Management, Massachusetts Institute of Technology.
- De Fazio, T. L., and Whitney, D. E. 1987. Simplified Generation of All Mechanical Assembly Sequences. *IEEE Journal of Robotics and Automation* 3: 640-658.
- Delchambre, A.; Coupez, D.; and Gaspart, P. 1989. Knowledge-Based Process Planning in Robotized Assembly. Presented at the SPIE Conference on Applications of Artificial Intelligence VII, Orlando, Fla., 27-31 March.
- Dixon, J. 1988. Designing with Features: Building Manufacturing Knowledge into More Intelligent CAD Systems. In Proceedings of Manufacturing International, 51-57. Fairfield, N.J.: American Society of Mechanical Engineers.
- Graves, S. C., and Holmes, C. H. 1988. Equipment Selection and Task Assignment for Multiproduct Assembly System Design. *International Journal of Flexible Manufacturing Systems* 1(1): 31-50.
- Gustavson, R. E. 1988. Design of Cost-Effective Assembly Systems. Pre-

sented at Successful Planning and Implementation of Flexible Assembly Systems, Ann Arbor, Mich., 29–31 March.

Homem de Mello, L. S. 1989. Task Sequence Planning for Robotic Assembly. Ph.D. diss., Robotics Institute, Carnegie Mellon Univ.

Homem de Mello, L. S. 1988. Automatic Generation of Mechanical Assembly Sequences, Technical Report, CMU-RI-TR-88-19, Carnegie-Mellon Univ.

Ko, H., and Lee, K. 1987. Automatic Assembly Procedure from Mating Conditions. *Computer-Aided Design* 19:3–10.

Kroll, E.; Lenz, E.; and Wolberg, J. 1989a. A Knowledge-Based Solution to the Design-for-Assembly Problem. *Manufacturing Review* 1(2): 104–108.

Kroll, E.; Lenz, E.; and Wolberg, J. 1989b. Rule-Based Generation of Exploded Views and Assembly Sequences. *Artificial Intelligence for Engineering, Design, Analysis, and Manufacturing* 3(3): 143–155.

Laszcz, J. F. 1984. Product Design for Robotic and Automatic Assembly. In Proceedings of the Robots 8 Conference, 6.1–6.22. New York: Society of Manufacturing Engineers.

Luby, S. C.; Dixon, J. R.; and Simmons, M. K. 1986. Designing with Features: Creating and Using a Features Database for Evaluation of Manufacturability of Castings. *Computers in Mechanical Engineering* 5(3): 25–33.

Lui, M-C. M. 1988. Generation and Evaluation of Mechanical Assembly Sequences Using the Liaison Sequence Method. Master's thesis, Mechanical Engineering Department, Massachusetts Institute of Technology.

Pappu, S. 1989. A Dual Descent Algorithm for Finding the Optimal Test Strategy for an Assembly Sequence. Master's thesis, Operations Research Center, Massachusetts Institute of Technology.

Phillips, R. E., and Aase, J. 1990. An Integrated Environment for Concurrent Engineering. In Proceedings of the Second National Symposium on Concurrent Engineering, 487–499. Morgantown, W. Va.: Concurrent Engineering Research Center.

Poli, C., and Fenoglio, F. 1987. Designing Parts for Automatic Assembly. *Machine Design* 59(29): 140–145.

Pratt, M. J., and Wilson, P. H. 1985. Requirements for Support of Form Features in a Solid Modeling System, Final Report R-85-ASPP-01, CAM-I, Inc., Arlington, Texas.

Sriram, D.; Logcher, R. D.; Groleau, N.; and Cherneff, J. 1989. DICE: An Object-Oriented Programming Environment for Cooperative Engi-

neering Design, Industrial Liaison Program Report, Massachusetts Institute of Technology.

Stix, G. 1991. Plane Geometry. *Scientific American* 264(3): 110-111.