SAND98-2759C

Advancing Design-for-Assembly
The Next Generation in Assembly Planning

Terri L. Calton

Intelligent Systems and Robotics Center
Sandia National Laboratories*
Albuquerque, NM 87185-1008

Abstract

At the 1995 IEEE Symposium on Assembly and Task Planning, Sandia National Laboratories introduced the Archimedes 2 Software Tool [2]. The system was described as a second-generation assembly planning system that allowed preliminary applications of assembly planning for industry, while solidly supporting further research in planning techniques. Sandia has worked closely with industry and academia over the last four years. The results of these working relationships have bridged a gap for the next generation in assembly planning. The goal of this paper is to share Sandia's technological advancements in assembly planning over the last four years and the impact these advancements have made on the manufacturing community.

1 Introduction

Manufacturing companies throughout the world are rapidly changing in order to survive in today's highly competitive market environments. Some examples of coping with changing environments are manufacturing globalization, automated and intelligent manufacturing, virtual manufacturing, and agile manufacturing. The objective of this movement in manufacturing is to improve flexibility, reliability and productivity, and to achieve competition-based technology development.

Accordingly, the main focus of Sandia's geometric reasoning research and development program is to provide intelligent software tools which automate many of the manufacturing processes that have traditionally been known to be the most costly, the most time-consuming, and the most error-prone. Some of these include part-level assembly planning, fixture planning, grasp planning, motion planning, tools planning and cost analysis. Sandia's overall strategy to reduce these costs is to push the breadth of application and depth of analysis and to find an appropriate balance between human and machine planning. Figure 1 helps illustrate this concept. The ultimate goal is to improve profitability of operations by developing smart software. The goal of this paper is to share Sandia's technological advancements in geometric reasoning capabilities and in assembly planning.

As with the Archimedes 2 system, Archimedes 4.0 can viewed as a sequence of modules each viewing the product at a greater level of detail and supplying more detailed assembly plans and designer feed back than the previous one; however, Archimedes 4.0 offers greater power and flexibility than its predecessor. The developers of the software focused on the limitations of the Archimedes 2 system and the needs of the manufacturing community to provide better solutions quicker.

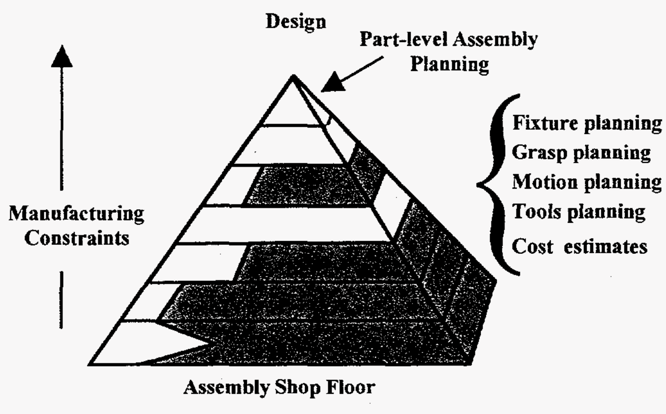

Figure 1. Geometric-reasoning for manufacturing processes.

A review of the limitations of the Archimedes 2 system is discussed in the next section. Following that, an overview of the Archimedes 4.0 system components is provided. This overview is used to provide solutions to Archimedes 2 limitations and to inform the readers of additional enhancements and features incorporated into the new system. A brief examination of the issues in lifecycle

*Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin company, for the United States Department of Energy under contract DE-AC04-94-AL85000.

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

engineering is also considered. In Section 4 the output capabilities of the system are discussed while Section 5 presents some experimental results of system applications. Finally, the paper concludes with a discussion of limitations specific to Archimedes 4 as well as providing future areas of research.

2 Archimedes 2 Review

The Archimedes 2 system was seen as a sequence of modules, each viewing the product at a greater level of

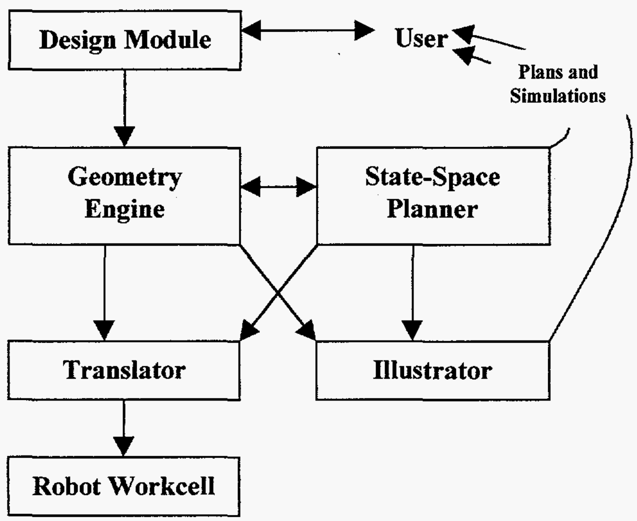

Figure 2. The architecture of Archimedes 2.

detail and supplying more detailed assembly plans and designer feedback than the previous one. At the top is the design module, which captures and represents the geometric, mechanical, and other information about the product required for analysis. The design module only required design consistency; it did not apply any manufacturing constraints. At the bottom is the robotic workcell; all details must by definition be present in the assembly plans executed there. The architecture of the system is shown in Figure 2. Using industry-standard languages for portability, maintainability, and compatibility with industrial users was a primary focus in writing Archimedes 2. Due to space limitations, the reader is referred to [2] for more detailed descriptions of the individual modules.

There were several specific areas of limitations to the Archimedes 2 system, most of them were and still are difficult issues not addressed adequately by any assembly planners to date. The Archimedes 4.0 system has made substantial progress in addressing those limitations. Some of the limitations included: lack of constraint representation (e.g., gripper design and grasp planning, fixture design, and motion planning), efficient search algorithms, inadequate facility for users to interact with the software, and a lack of non-geometric data representation.

3 Archimedes 4.0

The Archimedes 4.0 system is a constraint-based interactive assembly planning software tool used to plan, optimize, simulate, visualize, and document sequences of assembly [14]. Given a CAD model of the product, the program automatically finds part-to-part contacts, generates collision-free insertion motions, and chooses assembly order. The engineer specifies a quality metric in terms of application-specific costs for standard assembly process steps, such as part insertion, fastening, and subassembly inversion. Combined with an engineer's knowledge of application-specific assembly process requirements, Archimedes allows systematic exploration of the space of possible assembly sequences. The engineer uses a simple graphical interface to place constraints on the valid assembly sequences, such as defining subassemblies, requiring that certain parts be placed consecutively with or before other parts, declaring preferred directions, etc. The user interface is critical to effectiveness and user acceptance of an interactive planning system.

Archimedes 4.0 is implemented in C++ using ACIS® solid modeling kernel and Tcl/Tk for the graphical interface. The planner allows users to add product-specific assembly process constraints through the graphical user interface [13, 14]. Disassembly operations are generated using the NDBG approach discussed in [20]. Animation and user interface routines use OpenGL™ and X Windows™.

The system considers thousands of combinations of ordering and operation choices in its search for the best assembly sequences and ranks the valid sequences by the quality metric. Graphical visualization enables the engineer to easily identify process requirements to add as sequence constraints. Planning is fast, enabling an iterative constrain-plan-view-constrain cycle. For some restricted classes of products, it determines plans that optimize a given cost function, graphically illustrates those plans with simulated robots, and facilitates the generation of robotic programs to carry out those plans in a robotic workcell.

Figure 3 represents the overall structure of the system. At the top-middle and on the left-hand side are the design and constraint modules, which capture and represent the geometric, mechanical, and other information about the product required for analysis. These constraints come from a wide variety of sources: design requirements, part and tool accessibility, assembly line and workcell layout, requirements of special operations, and even supplier relationships can drive the choice of a feasible or preferred assembly sequence.

The modules listed on the right-hand side are the output modules. They include options to capture the sequences in the form of 3D-animations and videos, textual scripts and snap-shots that can be used for maintenance instructions and technical publications. The system also generates skeleton scripts to run robots, cost analysis information, and ergonomic analysis information.

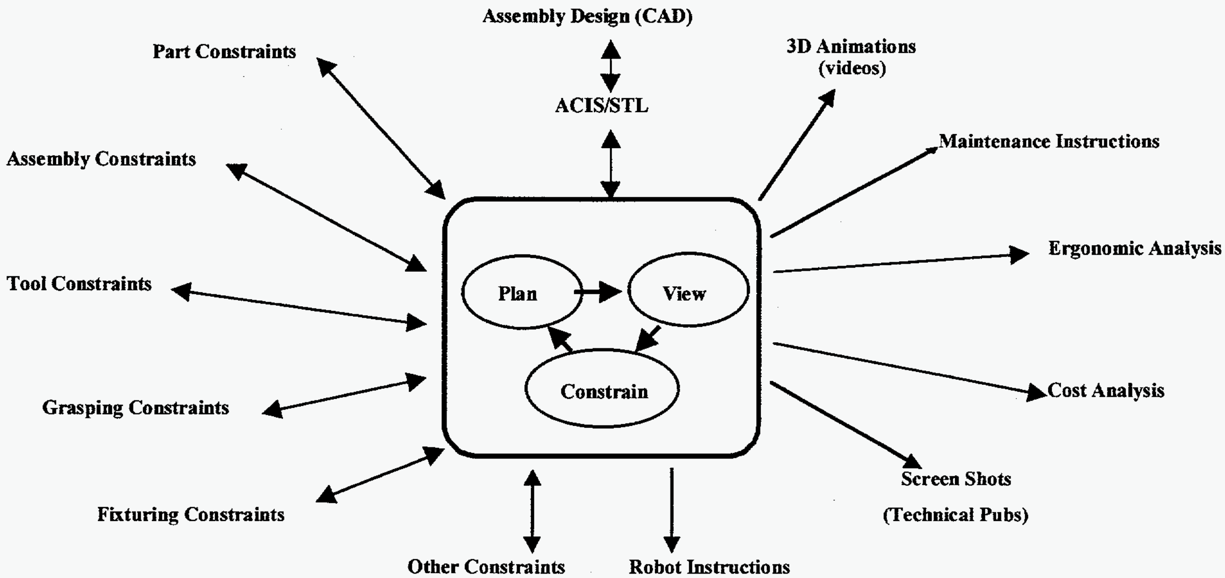

Figure 3. The Archimedes 4.0 Assembly Analysis and Planning Software System.

2.1 Design Module

The Archimedes 2 system was to some degree limited to a Pro/ENGINEER®-developed translation module which translated the geometric and auxiliary data into ACIS® for input into the downstream Archimedes planners. During the development of the current system, problems arose in the work with a variety of applications because of a lack of a standardized CAD data format. Since most CAD systems can produce STL or equivalently formatted data, the design module was extended with an STL-to-ACIS® translator. The STL format uses a straightforward facetted representation that sacrifices all of the higher-level geometric and topological information. STL data is, therefore, inferior to true solid modeling formats for Archimedes' purpose, except that it provides a way to import data into Archimedes that cannot be otherwise be translated.

2.2 Geometry Engine

The Archimedes 4.0 system implements the same geometric reasoning algorithms as its predecessor. The planner uses a non-directional blocking graph of each assembly to quickly identify important directions of motion and subassemblies [20]. A graphics workstation's hardware Z-buffer is used to quickly find collisions between complex facetted models.

2.3 State-space Planner

The search space implemented in the Archimedes 4.0 system is an AND/OR graph of subassembly states and the operations used to construct them from smaller subassemblies. The strategy is designed to generate a first plan as quickly as possible, like a depth-first search, but to avoid getting caught by bad early decisions as a depth-first search would. This is critical to achieve the desired view-constrain-replan cycle of interaction. During each pass of the search algorithm, a single assembly sequence is generated, making random choices of operations to construct each subassembly. The first time any subassembly is visited, only a single operation is generated to construct it, and the known subassemblies of that operation are then visited. Bounds on quality measures for each subassembly and operation are stored and propagated in the AND/OR graph as they are generated. This allows useless search paths to be identified and pruned, and an optimal plan to be identified when it becomes available. The same algorithm functions as an any-time algorithm to optimize the assembly sequence when the user requests.

2.4 Constraints Framework

The constraint framework provides a library of constraint types [13], from which the users can instantiate constraints on the assembly plan. This framework provides the underlying mechanics to optimization algorithms. Previous efforts to incorporate a comprehensive set of user constraints in assembly planners were based on liaison precedence relations that specified logical combinations of part connections that must be established either before or after others. Precedence relations were pioneered by Bourjault [3] and greatly extended by DeFazio and Whitney [10]. Wolter et al [22] analyze the expressive power of precedence relations in detail. Precedence relations are quite powerful, but they can be very difficult to write correctly or understand as a user of an assembly planner. A procedural approach over precedence relations was selected for reasons of efficiency and simplicity of implementation.

The Archimedes 4.0 system demonstrates that an assembly planning system can achieve comprehensive constraint coverage while maintaining the advantages of a procedural representation. The reader is referred to [14] for additional details on the constraint implementation.

2.4.1 Part and Assembly Constraints

Two types of constraints of assembly plans have been integrated into the system. Strategic constraints apply to the entire assembly and its plan, while tactical constraints only apply to certain subsets of the parts. Archimedes currently implements strategic constraints as flags for the planner. However, in theory there is no real difference, since tactical constraints can usually be applied to the entire assembly, and strategic constraints can always be limited to a subset of the parts.

2.4.2 Tool Constraints

Planning for assembly requires reasoning about various tools used by humans, robots, or other automation to manipulate, attach, and test parts and subassemblies. Constraints on assembly plans deriving from the need to use various tools in assembly or disassembly are called tool constraints. A framework to represent and reason about geometric accessibility issues for a wide variety of such assembly tools has been integrated into Archimedes 4.0. Central to the framework is a use-volume encoding a minimum space that must be free in an assembly state to apply a given tool, and placement constraints on where that volume must be placed relative to the parts on which the tool acts. Determining whether a tool can be applied in a given assembly state is then reduced to an instance of the FINDPLACE problem [16]. The reader is referred to [21] for a more complete analysis of the tools constraint framework.

2.4.3 Grasping Constraints

Two approaches to integrating grasping constraints into the Archimedes 4.0 system have been implemented. The first is an extension to the tools constraints framework described above. The second provides automatic selection and placement for suction and parallel-jaw grippers typically used in robotic assembly.

In the second approach, each gripper in the library is tested for feasibility during the generation of assembly sequences and ranked according to a user-specified metric. Suction grippers are optimally placed on components using the same direction of motion determined by the geometry engine. Ideally, the centroid of the object is selected; however, if the planar surface of the object does not accommodate the suction gripper, a spiraling technique is implemented to locate the nearest point to the center of gravity as possible. In the case parallel-jaw grippers, placement is determined automatically based on the component geometry. The algorithms check for placement of the jaws in the open-positions initially using a 360-degree rotation. The jaws are then checked for paths of closure.

2.4.4 Fixturing Constraints

A first attempt to include fixturing constraints into the Archimedes system is presented here. Most automated manufacturing, assembly, and inspection operations require fixtures to locate and hold parts. Given part shape and desired position and orientation, fixtures are usually custom designed by manufacturing engineers and machinists. In [4], a complete algorithm for designing modular fixtures for polygonal parts is presented. A class of modular fixtures that prevent a part from translating or rotating in the plane using four point contacts on the part's boundary was considered. These fixtures are based on three round locators; each centered on a lattice point, and one translating clamp. The algorithm accepts a polygonal part shape as input and constructs the set of all fixture designs that achieve form closure for the given part. The algorithm is guaranteed to find the optimal fixture, relative to any well-defined quality metric.

The algorithm presents example fixtures that are designed and includes a metric to rank fixtures based on their ability to resist applied forces. Recently these designs have been converted to ACIS® objects that can be directly imported into Archimedes. The system determines feasibility of assembly and generates an assembly plan with the fixtures declared as base-components via the constraint framework. Future work is aimed at linking the systems to produce both optimal fixtures and plans based on a common metric. Details describing this framework can be found in [9].

2.4.5 Other Constraints

As in initial assembly, the product design and known process constraints are inputs to creating such plans. However, for lifecycle assembly planning processes different goals and constraints, compared to initial assembly, require significant re-analysis of fundamental assumptions and methods underlying current assembly planning techniques. Some of those issues that require re-analysis include:

Disassembly operations: The mechanics of disassembly operations must be characterized as to feasibility and cost, and differ greatly from their assembly counterparts for planning purposes. For instance, pressfits are rarely possible to disassemble without damaging one or both parts, which is sometimes acceptable.

Partial assembly: Disassembly does not always proceed from or result in individual parts. For instance, a field upgrade may only require partial disassembly of a system to replace specified subassemblies.

Non-monotonic assembly: In the assembly planning literature, operations are non-monotonic when they leave parts in intermediate positions rather than placing or removing them completely. For instance, removing three screws from an access plate and leaving it hanging on the fourth screw is non-monotonic.

Destructive disassembly: In some applications, operations that destroy parts (i.e., cutting, tearing, or melting) are acceptable in disassembly.

There is little prior art on planning lifecycle assembly processes and the issues thus raised. Non-monotonic assembly planning is the most difficult issue computationally and is known to be PSPACE-hard [22], and the only system to generate such plans is limited to an impracticably small number of parts [12]. The only known study of planning methods for destructive disassembly uses a simplified model of destructive operations that does not correspond to what is seen in practice [11]. The commercially available ReStar system [18] attempts to optimize disassembly processes for recycling and is based on a service-assembly planner described in [19]. Both rely on user input to determine all possible operations, making them impractical on products of more than ten to twenty parts.

Related research outside assembly planning is more extensive. Programs from Boothroyd-Dewhurst enable design for service and recycling by analyzing plans entered by the user, but do no planning or optimization. Researchers in concurrent engineering and green engineering have studied design-for-service and design-for-disposal (for instance [1, 5, 15]), but lack of assembly planning capability limits them to heuristic and statistical methods.

Milner and Graves [17] developed a heuristic search through the multitudes of sequences to find those of nearly least-cost using simulated annealing (SA) to make such a search. However, a primary drawback of this system was that the least-cost sequences found by SA were often not of good engineering quality because engineering nuances could not be captured by the cost function.

Due to the inherent flexibility of the constraint system and the optimizing search algorithm, additional constraints have been employed in the system that address some of these drawbacks and limitations. In the Archimedes 4.0 system, the constraint-based assembly planning algorithms are combined with SA heuristics to produce optimal disassembly sequences. See [7] for further details.

3 Output Modules

As with the constraints, most of the output modules of the Archimedes 4.0 system are individual research areas. Due to space limitations, only brief descriptions and references are provided.

3.1 3D Animations and Screen Shots

While a user can readily visualize the assembly sequences using the graphical user interface, ancillary visual media are necessary to convey the results to other professionals in the manufacturing community whether it is the person making the business decisions or the person who is assigned to assemble the product. Output features integral to the system are the abilities to capture planned sequences in the form of 3D animations and screen shots of individual steps in the process. Videos demonstrate manufacturability and assist in training. Screen shots used in technical publications help reduce the effort required for generation of these documents.

3.2 Maintenance Instructions

Maintenance instructions are company specific and often time site-specific. To date, there does not exist an industry standard to characterize maintenance instructions. The plans generated using the Archimedes system provide some information for a company. Using this foundation, algorithmic modifications can be readily implemented to tailor the output formats to meet the company's needs.

3.3 Cost Analysis Information

One of the most critical forms of output is cost analysis information. A framework for cost optimization has been integrated into the system that can calculate costs, in dollars or other units, associated with the assembly or disassembly of a product. This is an extremely important feature. It provides a powerful tool for comparing costs of competing designs, upgrade vs. new product. Initial cost estimates are based on generic "handling" costs (e.g., the cost of an "insertion") and on the item (part) costs. As more information becomes available, the user may refine these costs and times by manually editing generic cost. After Archimedes has generated an assembly plan, cost and time estimates are obtained by selecting the Design-for-Lifecycle-Cost Analysis (DFLC Analysis) option from the File menu on the main Archimedes panel. The results are presented to the user in the form of a spreadsheet. Future work is aimed at providing optimization criterion on disassembly and assembly operations to minimize both dollars and time. Details describing this framework can be found in [8].

3.4 Ergonomic Analysis

Most assembly planning systems, including Archimedes 4.0, sometimes generate what appear to be perfectly valid remove-and-replace scenarios, which cannot physically be carried out by a human. For example, a screwdriver may be deemed feasible for an assembly operation; however, when a hand is placed on the tool the maintenance operation is no longer feasible, perhaps because of inaccessibility, insufficient strength, or human interference with assembly components. Similarly, human figure models may indicate that maintenance operations are not feasible and consequently force design modifications; however, if engineers had the capability to quickly generate alternative remove-and-replace scenarios, they might have identified a feasible solution. To solve this problem, a framework [6] for incorporating human models and humans factors (e.g., collision of the human with assembly components, human strength, energy and fatigue characteristics and time assembly motion) has been integrated into the system.

3.5 Robot Instructions

Archimedes 4.0 implements the same skeleton strategies for assembly operations that were implemented in the Archimedes 2 system to generate V+ robot instructions. The desired relative part locations and approach directions, as well as the sequence of mating operations, re-orientations, and welding operations, are derived from the input plan.

4 Experiments

Throughout the development the system has been applied to a wide variety of products from industry and government and has been tested on over 100 assemblies. Assembly part-count ranges from 5 to 1477. ACIS® data sizes range from 0.2 MB to 212 MB where the data for each distinct part is counted only once, regardless of the number of times that part appears in the assembly. Planning times vary from 4 seconds up to approximately 6 hours. Planning times given are to load in the pre-facetted data, identify all contacts in the assembly, and find a single geometrically valid part-level assembly sequence on an SGI Indigo Extreme workstation. Statistical results indicate savings in both time and money. Early reports by some users show more than a 75% reduction in time schedules, and a 25% reduction in prototype-fabrications cost.

5 Conclusion

The Archimedes 4.0 system has been briefly described. In [2], the authors discussed limitations in the Archimedes 2 system. While the Archimedes 4.0 system attempts to overcome these shortcomings, there are still a number of difficult issues not adequately addressed. A very pivotal point in the development of the software was capturing the designer's intent with the implementation of the graphical user interface and the constraint framework. The integration of the AND/OR search algorithm helped overcome the limitations inherent in the original A* search algorithm implemented in the Archimedes 2 system and the development of the STL-to-ACIS® converter broadened the breadth of application capabilities.

The software has been applied to numerous products. Often times, these applications have driven the research and development directions. In particular, the cost analysis module and the ergonomics modules are two areas of research resulting from an Archimedes needs workshop. Future work will proceed in the development of more integral and robust optimization algorithms in motion planning and grasp planning.

6 Acknowledgements

Over the years Archimedes team members have come and gone. The author wishes to thank all of the engineers and software developers who contributed to development of Archimedes 4.0. In addition to the authors of [2], Russell Brown, Ralph Peters, Glenn Laguna, and Emily Mitchell, all from the Intelligent Systems and Robotics Center at Sandia, are recognized for their contributions.

7 References

[1] D. Allen and T. R. Roose. Life cycle assessment and design for environment. In P. M. Eisenberger. Editor, Basic Research Needs for Environmentally Responsive Technologies of the Future: An Integrated Perspective of Academic, Industrial, and Government Researchers. Princeton Materials Institute, 1996.

[2] A. L. Ames, T. L. Calton, R. E. Jones, S. G. Kaufman, C. A. Laguna, and R. H. Wilson. Lessons learned from a second generation assembly planning system. In *Proc. IEEE Intl. Symposium on Assembly and Task Planning*, 1996.

[3] A. Bourjault. *Contribution à une aproche méthodologiuqe de l'assemblage automatisé: élaboration automatique des séquences opératories*. PhD thesis, Faculté des Sciences et des Techniques de l'Université de Franche-Comté, 1984.

[4] R. C. Brost and R. R. Peters. Automatic design of 3-d fixtures and assembly pallets. *In Proc. IEEE Intl. Conf. on Robotics and Automation*, pages 495-502, April 1996.

[5] D. S. Burks, M. D. Marks, and K. Ishii. Life cycle design for recyclability. In D. Navin-Chandra, editor, Green Engineering, Academic Materials Institute, 1996.

[6] T.L. Calton. A framework for geometric reasoning about human figures and factors in assembly operations. Submitted *to 3rd Intl. Conf. on Engineering Design and Automation*, 1999.

[7] T.L. Calton and R. G. Brown. An optimizing algorithm for automating lifecycle assembly processes. Submitted *to IEEE Intl. Conf. On Robotics and Automation*, 1999.

[8] T.L. Calton and R.R. Peters. A framework for automating cost estimates in assembly processes. Submitted *to IEEE Intl. Symposium on Assembly and Task Planning*, 1999.

[9] T.L. Calton and R.R. Peters. A practical approach for integrating automatically designed fixtures with automatic assembly planning. Submitted *to 3rd Intl. Conf. on Engineering Design and Automation*, 1999.

[10] T. L. De Fazio and E. E. Whitney. Simplified generation of all mechannical assembly sequences. In IEEE Journal of Robotics and Automation, RA-(6):640-658, 1987. Errata in RA-4(6):705-708.

[11] R. Gadh, Z. Ashai, and K. Lee. Computer-aided design-for-disassembly; as applied to design-for-environment. Technical Report I_CARVE-1091, Dept. ME, Univ. of Wisconsin, 1994.

[12] R. L. Hoffman. Automated assembly in a CSG domain. In Proc. IEEE Intl. Conf. On Robotics and Automation, pages 210, 215, 1989.

[13] R. E. Jones and R. H. Wilson. A survey of constraints in assembly planning. In Proc. IEEE Intl. Conf. On Robotics and Automation, pages 1525-32, 1996.

[14] R. E. Jones, R. H. Wilson, and T. L. Calton. Constraint-based interactive assembly planning. In IEEE *Intl. Conf. On Robotics and Automation*, pages 913-920, 1997.

[15] P. Langley. Systematic and nonsystematic search strategies. In *Artificial Intelligence Planning Systems: Proc. Of the First Intl. Conf.*,1992.

[16] T. Lozano-Perez. Spatial planning: A configuration space approach. In *IEEE Transaction on Computers*, C-32(2):108-120, 1983.

[17] J. M. Milner and S. C. Graves. Using simulated annealing to select least-cost assembly sequences. In IEEE Intl. Conf. On Robotics and Automation, pages 2058-2063, 1994.

[18] D. Navin-Chandra. ReStar: A design tool for environmental recovery analysis. *In Proc. Intl. Conf. On Engineering Design*, 1993.

[19] N. J. Nilsson. *Principles of Artificial Intelligence*. Springer-Verlag, 1980.

[20] R. H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. In *Artificial Intelligence*, 71(2):371-396, 1994.

[21] R. H. Wilson. Geometric reasoning about assembly tools. Technical Report SAND95-2423, Sandia National Labs, 1996. To appear in *Artificial Intelligence*.

[22] J. D. Wolter. On the automatic generation of plans for mechanical assembly. PhD thesis, Univ. of Michigan, 1988.