

## A Framework for Automating Cost Estimates in Assembly Processes

Terri L. Calton and Ralph R. Peters

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

RECEIVED

DEC 14 1998

OSTI

### Abstract

*When a product concept emerges, the manufacturing engineer is asked to sketch out a production strategy and estimate its cost. The engineer is given an initial product design, along with a schedule of expected production volumes. The engineer then determines the best approach to manufacturing the product, comparing a variety of alternative production strategies. The engineer must consider capital cost, operating cost, lead-time, and other issues in an attempt to maximize profits. After making these basic choices and sketching the design of overall production, the engineer produces estimates of the required capital, operating costs, and production capacity. This process may iterate as the product design is refined in order to improve its performance or manufacturability.*

*The focus of this paper is on the development of computer tools to aid manufacturing engineers in their decision-making processes. This computer software tool provides a framework in which accurate cost estimates can be seamlessly derived from design requirements at the start of any engineering project. The result is faster cycle times through first-pass success; lower life cycle cost due to requirements-driven design and accurate cost estimates derived early in the process.*

### 1 Introduction

Success in today's market demands rapid product development to meet customer expectations at a fair price. To do this, most companies must rethink their business strategies and revamp their product development environments. Products must be designed by integrated teams who work together to create new ways of doing business in order to produce better products faster and more efficiently. In order to achieve this goal, the integrated product development team needs information and tools that will facilitate flexibility, innovation, speed, accuracy, and process improvement. Computer-aided assembly planning software tools with cost-estimate capabilities will help

achieve this goal.

A framework for calculating cost estimates has been integrated into an automated assembly analysis software tool, known as Archimedes [4, 8, 9]. The system calculates cost, in dollars or other units, associated with the assembly or disassembly of a product. This is an extremely important feature in that it provides a powerful tool for comparing costs of competing designs, upgrade vs. new product, etc.

This paper describes the principles and implementation of a framework that supports an interactive system to plan, optimize, simulate, visualize, and document assembly and disassembly processes. The paper further describes an integrated tool, the Design-for-Life-Cycle Cost Module, in which accurate cost estimates can be seamlessly derived from design requirements at the start of any engineering project. The result is faster cycle times through first-pass success; lower life cycle cost due to requirements-driven design and accurate cost estimates derived early in the process. In this work, emphasis has been placed on automatic and computer-aided generation, optimization, simulation, documentation, and "design-for" feedback of lifecycle assembly plans derived from product CAD data. Section 2 presents a brief overview of the system with an introduction to the Archimedes 4.0 automated assembly analysis software tool. In Section 3 the cost module is introduced while Section 4 describes the implementation of the cost module. Section 5 describes experiments using and testing the planner. Finally, Section 6 concludes the paper and gives directions for future areas of work.

### 2 Assembly Planning Framework

The cost module is implemented in the Archimedes 4.0 assembly planning system. Archimedes 4.0 is a constraint-based interactive assembly planning software tool used to plan, optimize, simulate, visualize, and document sequences of assembly [4, 8]. 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,

\*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**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## **DISCLAIMER**

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

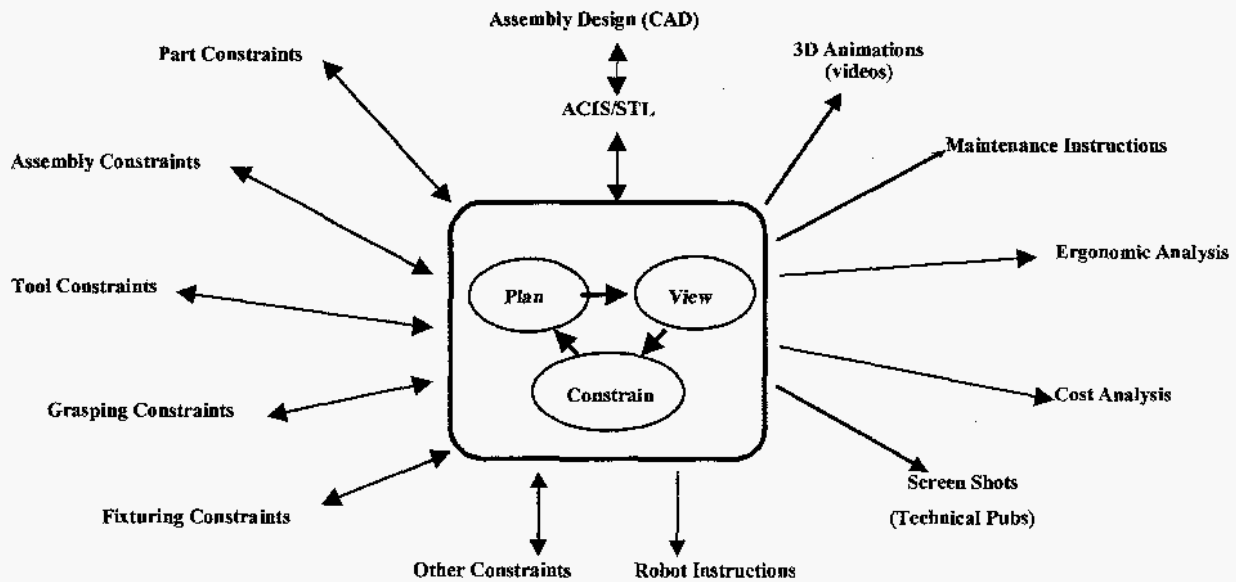


Figure 1. The Archimedes 4.0 Assembly Analysis and Planning Software System.

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. Archimedes 4.0 is implemented in C++ using ACIS<sup>®</sup> 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 [7, 8]. Disassembly operations are generated using the NDBG approach discussed in [13]. Animation and user interface routines use OpenGL<sup>™</sup> and X Windows<sup>™</sup>.

Figure 1 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; they can drive the choice of a feasible or preferred assembly sequence.

Those 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.

### 2.1 Constraint Framework

Constraint-based interaction has proven to be a powerful and intuitive paradigm for interactive assembly planning. Two types of constraints are implemented in the system. Strategic constraints apply to the entire assembly and its plan, while tactical constraints only apply to certain subsets of the parts. The system supports a framework [14] to represent and reason about geometric accessibility issues for a wide variety of such assembly tools. Two types of grasping constraints are also supported. The first is an extension to the tools constraint framework and the second provides automatic selection and placement for suction and parallel-jaw grippers typically used in robotic assembly. Fixturing constraints are also supported. Fixture design algorithms, [2], have recently been integrated with the Archimedes planner [6]. These algorithms accept polygonal part shape as input and construct the set of all fixture designs that achieve form closure for the given part. The designs are accepted as input to the assembly planning system, which determines feasibility of assembly and generates an assembly plan with the fixtures declared as base-components.

### 3 The Design-for-Life-Cycle Cost Module

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

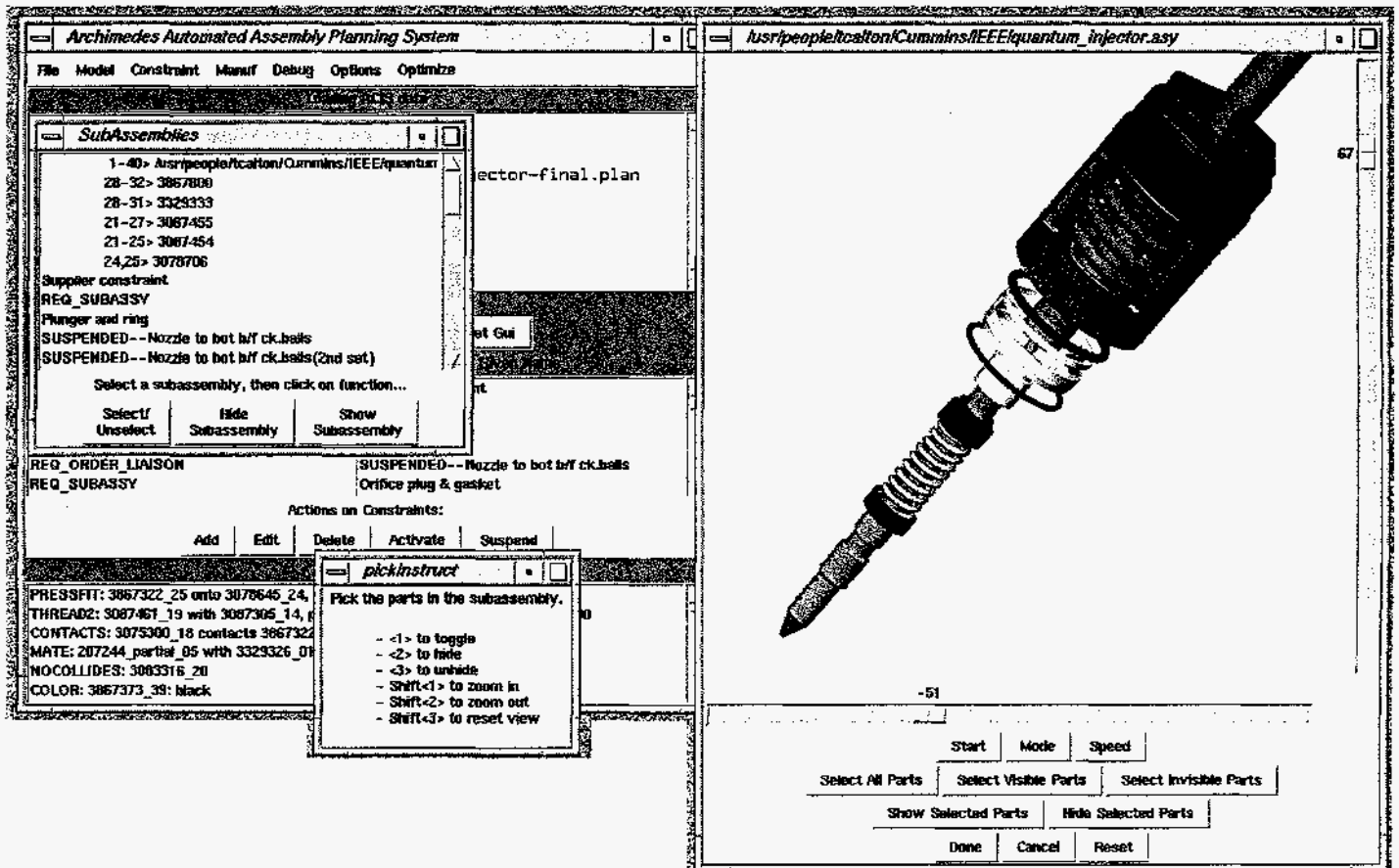


Figure 2. Archimedes 4.0 user interface. A subassembly constraint is being instantiated.

assembly, require significant re-analysis of fundamental assumptions and methods underlying current assembly planning techniques.

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, 3, 11]), but lack of assembly planning capability limits them to heuristic and statistical methods. Milner and Graves [10] developed a heuristic search through the multitudes of sequences to find those of nearly least-cost using simulated annealing (SA). 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. The inherent flexibility of Archimedes' optimizing search algorithms and constraint framework allow additional constraints to be added 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 [5].

### 3.1 Integrated Design to Cost

Software has been added to Archimedes 4.0 assembly planner that allows the user to easily estimate the cost and time associated with the assembly of a 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 files. After Archimedes has generated an assembly plan, cost and time estimates are calculated and presented to the user in the form of a spreadsheet. Costs for specific assembly steps may be edited to further refine cost estimates.

## 4 Interactivity and Implementation

The user interface is critical to effectiveness and user acceptance of an interactive planning system. Figure 2 shows the main Archimedes user interface. The left-hand-side of the window shows the program's current

status, displays any planning diagnostics, and allows pausing or aborting of computation. The right-hand-side of the window provides graphical output and part/subassembly selection and manipulation.

After loading the CAD data for an assembly and perhaps making some initial adjustments (geometric overrides [7]), the user selects "Plan". Constraints are added by clicking on the "Add" button located under the "Constraint" display, which brings up a sequence of menus and questions that let the user pick a constraint type and specify the particulars of the desired constraint. Once defined, constraints are listed in the planning dialog. The user can generate a plan and quickly view an animation of the sequence generated. The constraints can be edited using a process very similar to the initial definition. They can also be deleted, temporarily suspended, and re-activated. Constraint suspension is a very useful feature that allows the user to consider various scenarios for assembly. Constraints often embody assumptions about product assembly; by suspending some and replanning, the user can compare the cost of removing the assumption to the possible gains in assembly sequence efficiency that result.

After Archimedes has generated an assembly plan, cost and time estimates may be obtained by selecting the Design-for-Lifecycle-Cost Analysis (DFLC Analysis) option from the File menu on the main Archimedes panel. The DFLC analysis module generates an initial assembly process description. This is represented in a tabulated window (similar to a spreadsheet) as shown in Figure 3. Each row of the table corresponds to a different process step while the columns of the table describe different aspects of a step. The table initially describes a series of part (or subassembly) placement or removal operations listed in the order that the movements were generated by the Archimedes assembly planner algorithms.

Upon startup, the DFLC analysis software checks for three files containing cost and time information. The "specific" cost and time file contains information that is specific to these assembly operations. The "generic" handling cost file contains information that may be applied to assembly operations of this type. The item cost file contains cost information for each item or part. None of these files are required for use of the DFLC Analysis. However their existence, especially that of the generic handling cost file and item cost file, significantly simplify analysis for the user. The user may generate the generic handling cost and item cost files using a simple text editor or other costing software that contains the appropriate information. The specific cost file is usually generated within Archimedes.

A generic handling cost file consists of multiple instances of the following information. These files have names of the form "asm.gcst" where "asm" is the name of the assembly.

- Generic action (e.g., INSERT, INSERT WELD,

or REORIENT -- Archimedes defined)

- Comment (e.g., "Simple Insert" or "Insert with weld" -- user defined)
- Handling\_Cost (e.g., 1 -- user defined)

An item cost file consists of multiple instances of the following information and have a "icst" filename extension.

- Item name (e.g., 399426-u-a or ms51957-1; this name is based on the item's CAD name)
- Item cost (e.g., 1.23 -- user defined)

A specific cost and time file consists of multiple instances of the information listed below and have a "scst" filename extension.

- Specific action (e.g., INSERT THREAD2: ms51957-1\_15 with sh-r42886-000-u\_08 -- Archimedes defined)
- Comment (e.g., "Delicate insertion" -- user defined)
- Handling\_Cost (e.g., 1.00 -- user defined)
- Material\_Cost (e.g., 2 -- user defined)
- Step\_Time (e.g., 0.1 -- user defined)

Changes in entries for specific actions are usually made via the spreadsheet discussed below.

The DFLC analysis software assigns costs and times based on the information found in the cost files discussed above. First it looks in the specific-cost file for information. If no information is available for a particular assembly step, it estimates the handling cost based on information in the generic handling cost file, estimates the material cost based on information in the item cost file, and sets the time for that assembly step to zero.

After the initial costing is completed using the file information, a window opens and produces a table showing the individual assembly operations and process steps, handling cost, material cost, total cost, step time, and total time.

Associated with the spreadsheet is the graphics visualization window. A double-left click on the Step field moves the assembly animation to the start of that operation. Animation window controls can be used to

Step	Process/Operation	Handling	Material	Total Cost	Setup	Time
1009	INSERT: 3867414_35 onto 3867415_32	0.10	15.89	67.83	0.0	0.0
10100	INSERT: 3867419_37 onto 3867415_32 subassy	0.10	8.06	75.99	0.0	0.0
10110	INSERT: 3867418_assembled_40 onto 3867415_32 subassy	0.10	0.91	77.00	0.0	0.0
10120	INSERT: 3867419_38 onto 3867415_32 subassy	0.10	8.06	85.16	0.0	0.0
10130	INSERT: 3867373_39 onto 3867415_32 subassy	0.10	26.85	112.11	0.0	0.0
10140	INSERT: 3867415_32 subassy onto 3329326_01 subassy	0.10	0.00	112.21	0.0	0.0
10150	INSERT: 173486_partial_26 onto 3045047_partial_27	0.10	0.29	112.60	0.0	0.0
10160	INSERT: 173486_partial_26 subassy onto 3329348_21 subassy	0.10	0.00	112.70	0.0	0.0
10170	INSERT: 3076094_partial_11 onto 3329348_21 subassy	0.10	0.24	113.04	0.0	0.0
10180	INSERT: 3076096_partial_12 onto 3076094_partial_11 subassy	0.10	0.30	113.44	0.0	0.0

Step: 10140 | Simple Insert

Generic Cost file: /home/rrpeter/code/C++-code/archimedes/Assemblies/Cummins/Fuel\_Injector/IEEE/quantum\_injector.gen  
Item Cost file: /home/rrpeter/code/C++-code/archimedes/Assemblies/Cummins/Fuel\_Injector/IEEE/quantum\_injector.lst

Figure 3. Archimedes 4.0 generated cost analysis spreadsheet.

view that particular operation. Figures 3 and 4 demonstrate this concept. The process step associated with Step 10140 in the spreadsheet (Figure 3) is shown the graphics visualization window (Figure 4).

A double-left click on the Process/Operation field allows one to edit a comment attached to that particular operation. The comment associated with a particular assembly step is shown near the bottom of the window. One can edit the handling cost, material cost, and step time fields directly by double-left clicking on them. The total cost and total time fields are automatically updated.

Below the cost/time table is the comment for the selected assembly step. Immediately below the comment, is a listing of the files used for costing the assembly; zero to three files may be listed. The information shown in Figure 3 is estimating costs and times using only a generic handling cost file and an item cost file.

The three bottom buttons of the table allow input/output operations. Information contained in the table can be saved for later use with the "Save time/cost information" button; its output is a "specific cost and time file" which was previously discussed. The "Load time/cost information" allows one to use another file for specific cost and time information and overwrites what is currently in the table. Finally, the "Print time/cost information" outputs the table data to a file, which can be printed for later use.

### 5 Experimentation

The Archimedes 4.0 system has been briefly described. 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

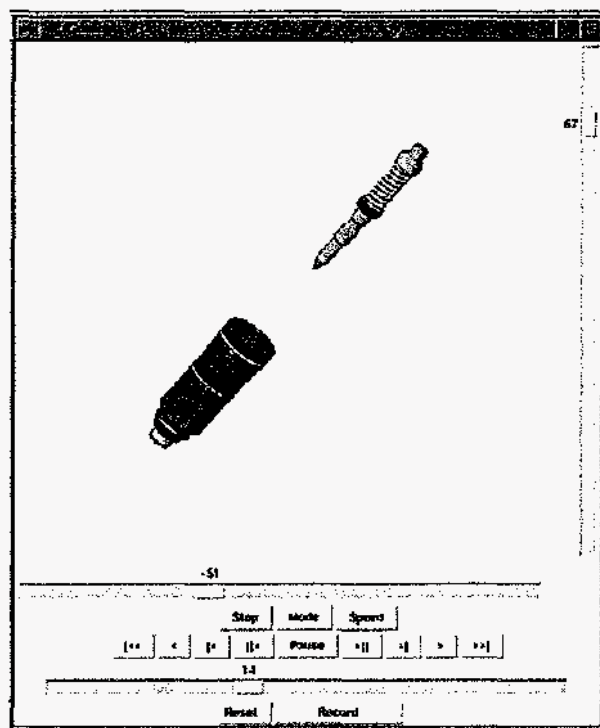


Figure 4. Archimedes 4.0 frame-grab to associate with Step 1080 in Figure 3.

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 those required to load in

the pre-faceted data, identify all contacts in the assembly, and find a single geometrically valid part-level assembly sequence. Statistical results indicate huge 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 costs.

However, due to the newness of the integrated cost module, application and testing of the cost module itself has been limited. The Fuel Systems Department at Cummins Engine Company in Columbus, Indiana has been very instrumental in the development of the cost module and its integration with the Archimedes planner. Cummins has assisted the development by providing suggestions of display formats and by providing real data<sup>1</sup> for testing purposes. Cummins also received a test-and-evaluation copy of the software to help guide future developments.

## 6 Conclusion

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 was just one area of research resulting from a recently held Archimedes needs workshop. The focus of this paper has been on the development of computer tools to aid manufacturing engineers in their decision-making processes. The framework described provides a tool in which accurate cost estimates can be seamlessly derived from design requirements at the start of any engineering project. The result is faster cycle times through first-pass success, lower life cycle cost due to requirements driven design and accurate cost estimates derived early on in the process. As mentioned earlier, constraint suspension is a very useful feature that allows the user to consider various scenarios for assembly. Constraints often embody assumptions about product assembly; by suspending some and replanning, the user can compare the cost of removing the assumption to the possible gains in assembly sequence efficiency that result. Future work is aimed at providing optimization criterion on disassembly and assembly operations to minimize both dollars and time.

## 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] 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.
- [3] D. S. Burks, M. D. Marks, and K. Ishii. Life cycle design for recyclability., In *Green Engineering, Academic Materials Institute, D. Navin-Chandra, editor 1996*.
- [4] T.L. Calton. Advancing design-for-assembly: the next generation in assembly planning. Submitted to *IEEE Intl. Symposium on Assembly and Task Planning, 1999*.
- [5] 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*.
- [6] T.L. Calton and R.R. Peters. A practical approach for integrating automatically designed fixtures with automatic assembly planning. Submitted to *IEEE Intl. Symposium on Assembly and Task Planning, 1999*.
- [7] 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.
- [8] 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.
- [9] S.G. Kaufman, R.H. Wilson, R.E. Jones, T.L. Calton, and A.L. Ames. The Archimedes 2 mechanical assembly planning system. In *Proc, IEEE Intl. Conf. On Robotics and Automation*, pages 3361-3368, 1996.
- [10] J. M. Milner and S. C. Graves. Using simulated annealing to select least-cost assembly sequences. In *Proc., IEEE Intl. Conf. On Robotics and Automation*, pages 2058-2063, 1994.
- [11] D. Navin-Chandra. ReStar: A design tool for environmental recovery analysis. In *Proc. Intl. Conf. On Engineering Design, 1993*.
- [12] R. H. Wilson. Minimizing user queries in interactive assembly planning. In *IEEE Trans. On Robotics and Automation*, 11(2):308-312, 1995.
- [13] R. H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. In *Artificial Intelligence*, 71(2):371-396, 1994.
- [14] R. H. Wilson. Geometric reasoning about assembly tools. Technical Report SAND95-2423, Sandia National Labs, 1996. To appear in *Artificial Intelligence*.

<sup>1</sup>Actual data was provided by Cummins and used during testing and benchmarking; however, for disclosure purposes the numbers shown have been altered.