7-14-95

SANDIA REPORT

SAND95-1340 • UC-705

Unlimited Release

Printed June 1995

The Automated Assembly Team Contributions to the APRIMED Agile Manufacturing Project

Rondall E. Jones, Arlo L. Ames, Terri L. Calton, Stephen G. Kaufman, Catherine A. Laguna, F. Michael Saavedra, Randall H. Wilson, Robert O. Woods

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: 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, nor any of their contractors, subcontractors, or their employees, makes 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, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

DISCLAIMER

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

The Automated Assembly Team Contributions to the APRIMED Agile Manufacturing Project

Rondall E. Jones Arlo L. Ames Terri L. Calton Stephen G. Kaufman
Catherine A. Laguna F. Michael Saavedra Randall H. Wilson
Robert O. Woods

Intelligent Systems and Robotics Center
Sandia National Laboratories
Albuquerque, NM 87185-0951

Abstract

The Automated Assembly Team of the APRIMED¹ Project (abbreviated as *A'* herein) consists of two parts: the Archimedes Project, which is an ongoing project developing automated assembly technology, and the *A'* Robot Team. Archimedes is a second generation assembly planning system that both provides a general high-level assembly sequencing capability and, for a smaller class of products, facilitates automatic programming of a robotic workcell to assemble them. The *A'* robot team designed, developed, and implemented a flexible robot workcell which served as the automated factory of the *A'* project.

In this document we briefly describe the role of automated assembly planning in agile manufacturing, and specifically describe the contributions of the Archimedes project and the *A'* robot team to the *A'* project. We introduce the concepts of the Archimedes automated assembly planning project, and discuss the enhancements to Archimedes which were developed in response to the needs of the *A'* project. We also present the work of the *A'* robot team in designing and developing the *A'* robot workcell, including all tooling and programming to support assembly of the *A'* discriminator devices. Finally, we discuss the process changes which these technologies have enabled in the *A'* project.

¹APRIMED stands for Agile Product Realization of Innovative ElectroMEchanical Devices, a Sandia National Laboratories project that was active from the summer of 1993 until March 31, 1995. The acronym APRIMED is often written with the mathematical "prime symbol," as *A'*.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

1 Introduction

1.1 The APRIMED Project

In June 1993 Sandia National Laboratories initiated an agile manufacturing project named Agile Product Realization for Innovative Electro-Mechanical Devices (or APRIMED, which is often abbreviated with the mathematical prime symbol as A'). The stated goal of the A' project was to develop a much faster design-to-production cycle for quality electromechanical devices. The quality goal was for custom electro-mechanical devices to meet functional acceptance criteria first time, off-the-line, without the fine tuning typically necessary for such custom mechanisms.

An electro-mechanical safety device, the pin-in-maze discriminator, was chosen as the example family of products for A' product realization. This family of devices allowed A' to test its technologies by building precision devices for which there exist a wide range of potential customer applications. Such discriminators were originally developed for nuclear weapon applications to prevent detonation in abnormal environments. When the correct signal is applied to the discriminator, stepper motors successfully navigate a pin through a maze which contains a sequence of left and right turns. If an incorrect signal is applied, the pin enters a dead-end in the maze and the device locks up, preventing the weapon from being enabled.

The intent of the A' project was to develop, integrate, and demonstrate an extensible slice of an agile product realization enterprise. The project scope did not encompass all elements of an agile product realization enterprise, but focused on a complete design to production process emphasizing model-based design and production, computer simulation, and automation.

A' project team members were drawn from 20 different organizations at Sandia, and were grouped into seven teams: management, design, manufacturing, automated assembly, communications, analysis, and quality assurance. Various aspects of the project either have been documented or are in process of being documented. A report by Forsythe *et al*[3] discusses the overall product realization process, and the paper by Diegert *et al*[2] discusses the key concept of parameter space qualification for a Parent/Child familyTM. The present document presents the work of one of the seven teams, namely the automated assembly team. This team is best understood as consisting of two closely related subteams: the Archimedes project and the A' robot team.

1.2 The Role of the Automated Assembly Team in the APRIMED Project

Modern electromechanical products tend to have increasing complexity, with more parts, of diverse types, assembled into less and less space. These designs are created faster than ever before, often by geographically separated designers. Ensuring that these products can be assembled, serviced, and disassembled efficiently and reliably is a growing challenge to designers and manufacturing engineers.

Computer-aided assembly planning techniques promise to address these problems by providing engineers with software tools to automatically analyze assemblability, systematically explore alternative assembly schemes, and facilitate the design-to-manufacturing transfer. Responding to this need, the last decade has seen an explosion in computer-aided assembly planning research (see, e.g.,[4]). Significant advances have been made in both theory and practice, and a great number of experimental systems have been built.

The Archimedes project is a Sandia National Laboratories project to research automated assembly planning. The Archimedes project was originally started in the late 1980's as a prototype to prove the concept of vertically integrating the assembly planning process by starting with a CAD model and automating virtually the entire assembly planning process including the programming of a robot workcell to assemble that device. The project was reconstituted in 1992 with the goal of upgrading the prototypical effort to a production-like effort. A major milestone in achieving this goal was reached in the fall of 1994 with the demonstration of the Archimedes 2 system. This system is discussed in Section 2 below.

The Archimedes 2 system has been extended for application to the A' discriminator as a mutually beneficial arrangement: (a) Archimedes' technology was used to generate assembly plans for the discriminator and simulate them, and workcell programming technology from Archimedes 2 was applied to programming the A' workcell, whereas (b) A' provided a challenging real-world example to stretch the Archimedes project's ability to provide vertically integrated solutions (i.e, CAD model to assembly robot technology). These extensions for A' will be discussed at some length in Section 3.

Automated assembly planning would lack grounding without a workcell (or factory) in which the plans can be executed. It was the role of the A' robot team to design and implement such a cell. The task given to the team was to design a workcell with one main manipulator and an in-cell laser welder facility, and to provide maximum flexibility in the final application. In fact, high-level design of the cell was done before the type of discriminator device to be used was selected. (That is, before the "parent" in the Parent/ChildTM family was determined[2].) Of course, actual end-of-arm tooling and related details had to be implemented later in the process. Extensive use of vision was implemented to maximize flexibility and precision of operations. This topic will be discussed in Section 4 below.

Inclusion of these automated assembly aspects into the A' project from its inception gave rise to valuable collaborations and interesting process improvements. These process improvements range from the availability to designers of design-for-assembly software to a variety of inter-team collaborations which proved very valuable for resolving problems and shortening project time scales, thus contributing to agility in the product realization process. These will be discussed in Section 5.

2 Overview of Archimedes 2

2.1 Introduction

During the first year of the *A'* project, the Archimedes project was involved in developing and integrating the project's technologies into a vertically integrated, production-like system, which was named Archimedes 2 (after the prototypical integrated assembly planning system developed by David Strip and others in the late 1980's[7]). We present here a brief overview of Archimedes 2. Interested readers may see related papers[10] for further information on this system.

The Archimedes 2 system, which is written largely in C and C++ for efficiency and compatibility with commercial practice, quickly and automatically generates geometrically-valid assembly sequences for a wide range of assemblies, given input from industry-standard Computer-Aided Design (CAD) packages. For a more restricted class 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.

The Archimedes 2 system can be seen as a sequence of modules, each viewing the product at a greater level of detail and supplying more detailed assembly plans or designer feedback than the previous one. The architecture of the system is shown in Figure 1. 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 enables (but does not enforce) creation of a consistent design; it does 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.

Using industry-standard languages for portability, maintainability, and compatibility with industrial users was a primary focus in constructing Archimedes 2. The design module is written in Pro/DEVELOP®, and integrated with the Pro/ENGINEER®CAD package (usually referred to as Pro/E herein), while the Illustrator is implemented in CimStation®using SIL. The rest of the modules are written in C++, using ACIS®as a solid modeling kernel and GL®for on-line animated output.

2.2 Design Input

The Archimedes team chose ACIS as its geometric modelling software, due its open programmer's interface and its support of the geometric computations we need. On the other hand, Pro/E has been chosen by Sandia's design organizations as their preferred mechanical modelling software, due to the nature of its user interface. Thus, in order to provide access to the bulk of significant CAD models at Sandia, we had to develop an appropriate translation tool from Pro/E to ACIS. We built the translator on a part translation capability developed by Spatial Technologies, and extended it to support assemblies and the weld representation described below.

The Archimedes project does not intend to restrict its attention to models developed in

Figure 1: The architecture of Archimedes 2

Pro/E. Rather, as time permits we will obtain or develop equivalent translation tools for other CAD systems.

In order for an assembly planning process to be agile, a fundamental requirement is that a complete description of the assembly be directly provided to the assembly planning activity. Current CAD systems describe part shape and relative position, but generally do not note how parts are joined to each other (e.g. weld, glue, snap fit), and how they relate to each other in the final assembly (e.g. rotational freedom provided by a bearing, or spring loadings). We extended Pro/E to model weld data, which has historically been given on engineering drawings (e.g., weld on this curve, 4 spots), and added code to the Archimedes geometric planner to automatically infer a direction from which to weld. Additional extensions to cover the spectrum of part-part relationships required for mechanical assemblies is underway.

2.3 Geometry Engine

The geometry engine accepts ACIS assembly models and auxiliary data from the translator just discussed, and determines geometrically valid *part-level* or subassembly-level assembly sequences. These sequences consider only the geometric blocking constraints between the parts and/or subassemblies to be mated at each step. The geometry engine finds part-part

contacts automatically from the CAD data, then constructs a *non-directional blocking graph* of the assembly[11], to quickly identify important directions of motion and subassemblies. Single translations and twisting motions are used to mate parts. A graphics workstation's hardware Z-buffer is used to quickly find collisions between complex models.

Geometrically valid assembly sequences can be found by using a simple heuristic to select one of the valid mating choices at each step.

2.4 Optimizing Planner

This module calls the geometry engine to find geometrically valid part motions, but adds additional constraints in its search for an optimal assembly plan. It only allows vertical assembly, adding subassembly reorientation operations when required, and enforces laser weld-site accessibility. The planner performs an A* search[6] over the space of linear assembly plans (i.e. each operation places a single part in a subassembly) to find the best plan according to a user-specified optimality criterion. (For example, the criterion might be minimized tool changes, minimized subassembly inversions, minimized number of fixtures, or a combination of these.) The optimizing planner also suggests simple assembly fixtures designed using boolean subtraction. The resulting assembly plan can be output to the Illustrator or the workcell.

2.5 Illustrator

The Illustrator simulates assembly plans in robotic workcells. After the robotic workcell is modeled in CimStation, the data that must be supplied to the Illustrator for each simulation are:

- the CAD model of the assembly,
- an assembly plan generated by either of the above two methods,
- optionally, fixtures, which can be included as parts in the CAD model or modeled directly in CimStation.

The output is then a detailed simulation of the robot executing the plan. Heuristics are used to fill in missing plan details such as gripping points on the parts and part feeding locations. The resulting plan illustration allows workcell designers as well as assembly designers to visualize the assembly workcell in operation. Thus, newly generated assembly sequences from the geometric and optimizing planners can be immediately visualized in a simulated robotic environment.

Details of this process, such as expanding the Illustrator's "state machine" to include workcell actions not previously programmed, are discussed in some detail in Section 3, and in other reports[5]. While not discussed in detail here, the Illustrator also provides some "flying parts"-type simulations, including one in which a user-specifiable number of parts migrate to their proper locations simultaneously.

2.6 Robot Code Generation

The Archimedes translator compiles assembly plans into V+[®] code to execute in the Archimedes robotic workcell. This translator is the only software originally developed for the prototypical system[7] which is still being used. This translator takes as input a narrative description of the assembly, which is written by the optimizing planner. Required details such as geometry and locations of fixtures and part gripping offsets are retrieved from a workcell-specific database. Desired relative part locations and approach directions, as well as the sequence of mating operations, reorientations, and welding operations, are derived from the input plan. The workcell program uses a library of task-level robotic routines written in V+ to identify, localize, acquire, orient, and mate parts and subassemblies.

During development of Archimedes 2, new V+ routines were created which provided general feature recognition capabilities. These routines were built using AdeptVision AGS. As part of this effort, kinematic modules were developed to accommodate vision guidance. Then, part-specific modules were developed to accommodate the vision requirements in assembling the pattern wheel example.

2.7 Application

The design module and geometry engine are capable of operating on a large class of electromechanical assemblies and have been applied to a number of designs obtained from industrial sources. The optimizing planner and translator in Archimedes 2 are targeted toward creating detailed linear assembly plans for a more limited set of unidirectional assemblies. The Illustrator lies in-between: it is capable of illustrating any plan created by the geometry engine or optimizing planner, but the detail and realism are greater for unidirectional assemblies.

Interested readers can retrieve pictures of example assemblies, animations of sequences, Illustrator output, and video of the robotic workcell on the World-Wide Web at

<http://www.sandia.gov/agil/pg/assembly.html>

The example application which was used to demonstrate Archimedes 2 was a thirteen part subassembly from an older discriminator design. All the features discussed above were exercised on this example in September 1994, for a full vertically integrated demonstration from CAD model to assembled device. The workcell used to do this demonstration was an older Archimedes development workcell centered around an AdeptTwo robot. This system has two manipulator-mounted cameras for part localization and recognition; a tool changer that allows use of multiple grippers, including suction and parallel-jaw types; and a specialized part-inverting device. All welds were simulated since no in-cell welder was available.

This baseline system was extended to meet the requirements of the A' project, with the A' workcell as the target execution system. These extensions, and the development of the A' assembly workcell, are discussed in the following two sections.

Figure 2: The discriminator

3 Archimedes Software Development Motivated by A'

Following the demonstration of the Archimedes 2 baseline, development activities in the Archimedes project for the following half year were coordinated so as to extend the project's capabilities along the lines needed by the A' project, and to further the capabilities of the Archimedes project's main goals at the same time.

An exploded diagram of an example "child" design from A' is shown in Figure 2. This discriminator² was invented by Gilbert Benavides, who was influenced by exposure to the earlier prototypical Archimedes system[7], which highlighted the desirability of unidirectional assembly. This is a very complex device for any existing automated assembly planning system.

More than any other technology area in A' , the Archimedes project was in the mode of actually *developing* the technology during the project rather than applying, adapting, integrating, and communicating between existing technologies. The demonstration of the Archimedes 2 system was a major milestone in this development. Following that demonstration several aspects of the Archimedes system were extended with the specific intent of building a follow-on Archimedes system which could handle the A' discriminator build in the A' workcell. Of particular impact was the need to extend Archimedes 2 to handle subassemblies explicitly. This was done in order to reduce the intractable amount of time required by the optimizing planner in the absence of such guidance, and to support simulations of such subassembly oriented sequences. Adding this capability would not necessarily have been a priority for the Archimedes project except for the needs of the A' project. However, it is also an important capability to have for future activities of the Archimedes project.

The term "subassembly" has various connotations, which we need to clarify to avoid

²A discriminator is a safety device which prevents accidental firing of a weapon system by providing an open circuit in the arming and fusing system which is not closed until a precise sequence of electrical signals is given to the device.

confusion:

- Designer subassemblies. These are subassemblies defined by the designer, for any of a variety of purposes. Such subassembly information is recorded in the CAD files and can be extracted by users of the CAD data such as the Archimedes system. Not all such subassembly data is relevant to assembly planning, though, because designers may group objects for reasons not related to assembly concerns. For example, a designer might for some reason group all the bolts holding a lid in place as a subassembly, but that subassembly would not be usable information to the assembly planning system unless it at least also contained the parts being affixed.
- Recommended subassemblies. To be useful to the assembly planning system, subassemblies need to be defined as groups of parts that can be assembled into a unit that can then be treated as a single part. The Archimedes system can treat such subassembly descriptions as high level suggestions for ways to subdivide the assembly planning process.
- Essential subassemblies. Sometimes in the assembly planning process the planner will discover that certain parts logically *must* be removed/inserted as a group. These groups would then be treated as a subassembly.

Each subsection below will discuss extensions for subassemblies which were implemented for A' .

3.1 Advancement of the Archimedes Geometric Assembly Planner

While the Archimedes geometric planner was written as part of the Archimedes project rather than directly as part of the A' project, the A' discriminator is the type of product it was designed to handle: many complex parts, requiring the explicit use of subassemblies to construct it. As such, the discriminator became a very useful test case for the geometric planner technology. In addition, the slowness of the planning process when applied to the complex models in the discriminator drove a revolutionary technical advance in the way the planner finds collisions.

The geometric planner is a reimplementaion of the methods of [9] in an industry-standard computer language (C++), using a supported and industry-compatible CAD model representation (the ACIS geometric modeling kernel). The reimplementaion process provided an excellent chance to refine the code and build in support for future research and development efforts in assembly planning.

The geometric planner determines assembly operations (and sequences of operations) that are geometrically valid, i.e. those in which parts do not collide with each other. These operations and sequences are found in polynomial time using a structure called a *non-directional blocking graph*, or *NDBG*[11]. The NDBG specifically allows fast generation of subassemblies and their mating motions. We will not describe the NDBG further here; interested readers are referred to[11].

(Note: For those to whom the subject of automated assembly planning is unfamiliar, it is important to point out that virtually all algorithmic assembly planning approaches – including ours – employ goal-directed planning, starting with the completed design and determining what the *last* assembly action is. The process continues recursively until completion. Thus, the last assembly action determined by the algorithm is the first to be performed in the actual assembly process. This strategy allows one to take full advantage of all the geometric information which is explicitly and implicitly present in the CAD model of the assembly.)

The NDBG method implemented in the geometric planner determines possible assembly operations using part-part contacts; the resulting part motions must be checked for feasibility using a collision-detection routine. Initially, the planner used the ACIS sweeping and intersection capabilities to determine collisions along translational trajectories. Even when highly optimized, these routines required on the order of a second to several minutes to determine feasibility of a single part-part collision query for the discriminator, since the parts were so complex (more than 4MB of CAD data). Assembly planning requires thousands of such queries to be answered, which meant hours or even days of time to find an assembly plan for the discriminator.

A new method of collision detection was developed which exploits the hardware capabilities of a graphics workstation to answer collision queries for even the most complex discriminator parts in under a second, for single translational trajectories. The method is raster-based, discretizing the front and back silhouettes of the parts in a plane perpendicular to the direction of motion, but otherwise making no simplifying assumptions or approximations. The result was an amazing speedup in planning times—on the order of a thousandfold—so that applying the geometric planner to the discriminator now requires less than 30 seconds. A technical advance disclosure has been filed on the method, and a patent application is being considered.

The NDBG data structure used in the geometric planner is specifically designed for computing possible assembly motions for subassemblies, so to support use of recommended subassemblies all that had to be done was to tie the geometric planner to the Pro/E subassembly representation. Since the 40+ part count of the A' discriminator is well within the practical capacity of the current version of the geometric planner, and essential subassemblies were already handled correctly, there was no fundamental advantage to the geometric planner gained by using recommended subassemblies.

While not officially part of the A' project, the development of the Archimedes geometric planner is an excellent example of cooperation and leveraging: the A' application gave focus to and drove the creation of the geometric planner, while the resulting fast assembly planning capabilities are an integral piece of the agile assembly side of A' .

3.2 The Archimedes Optimizing Planner

The Archimedes optimizing planner is a somewhat higher level planner than the geometric planner which takes advantage of some of the geometric planner's capabilities. This planner evolved significantly over the lifetime of the A' project, with the single largest advance being

the extension to use subassembly advice.

The optimizing planner considers many possible assembly sequences, and finds a sequence of lowest cost according to a given cost function. (The particular cost function used in the A' application will be described below.) The planner is based on an algorithm that searches a tree structure of partial assembly decompositions of the given assembly. Each path through the tree represents a different partial assembly sequence; at each search step, the current best partial sequence is decomposed further. This process repeats until the assembly decomposition is complete. The search algorithm used is an implementation of the A^* search algorithm [6], which is guaranteed to find a path of minimum cost under certain assumptions.

Input to the planner is an ACIS model of the part geometry (in the assembled configuration); declared subassemblies; and weld locations. Output is a list of actions required to perform the assembly.

Before describing the implementation of the search algorithm, assumptions made about possible workcell actions must be described. The planner must respect the capabilities of the target workcell, so that it does not produce a plan that is impossible to execute. The current version of the planner (which we expect to adapt or modify for a variety of applications in the future) makes these assumptions:

1. We assume that subassemblies may be in one of two orientations, normal and inverted. One orientation is obtained from the other by rotating the subassembly 180 degrees about the x axis. We presume that there is a device in the workcell that can perform this inversion operation.
2. We assume that all insertion operations are translations in the downward (z) direction. This is a simplification which is consistent with typical usage of SCARA robots and which is not generally very limiting.
3. We assume that there are sufficient tools in the workcell to manipulate each part; the mapping between parts and the tools needed to manipulate them is given in a table. In particular, we assume that there is a laser welder in the workcell at a fixed position. (In reality, the laser's position is adjustable by the first two stages of the XYZ table it is mounted on, but this assumption is a reasonable simplification.)

Bearing these assumptions in mind, the search algorithm is described in three steps:

- the representation of a state;
- the method used to generate successors of a state in the search tree; and
- the cost function used to rate the desirability of a particular state.

A state is a data structure representing the parts that have not yet been decomposed during the planning process, the current orientation of the subassembly consisting of these unremoved parts, and the number of tool changes and subassembly inversions needed to

assemble the previously processed parts. (These last two values are functions of the path to the state, not the state itself). The state also includes information about the part or subassembly last removed.

The successors of a state are those states obtained by geometrically feasible part or subassembly decomposition, as determined by the geometric planner discussed in the previous section. Any declared subassembly that is a subset of the remaining parts of the current state is a candidate for decomposition, as is any remaining part not included in a declared subassembly. For any such part or subassembly decomposition operation to be feasible, it must pass several tests. These are:

1. The proposed state has not been generated before. (This is important because in general a given assembly state could be reached in a number of ways.)
2. The part or subassembly does not collide with any other remaining parts when removed.
3. Any weld sites are visible to the laser welder.

If a state fails any of these tests, it is inverted and the tests are reapplied. This is the only way for inversions to be introduced into the plan.

The collision checking is implemented by the geometry facilities discussed in the previous section. If the proposed removal operation does not have any collisions, any welds between the removed part or subassembly and any other remaining part are found. Each must be visible to the position of the laser welder. This test is implemented using the ACIS ray-firing facility. A ray is fired from an assumed, artificial welder position to the weld site, and the first object it hits is computed. The part hit must be one of the two parts joined by the weld, and the distance between the hit and the weld site must be less than a user-settable maximum.

If the proposed state passes these tests, it is deemed to be a successor to the given state. The number of tool changes and the number of inversions are updated, by comparing the tool needed for the removal to the the previously used tool, and comparing the orientation of the successor state to that of the given state. If either comparison fails, the respective count is incremented by one. The cost of the state is then computed as the sum of these two values, and the number of parts not yet removed.

When all successor states have been generated, the lowest cost state must be selected. To do this efficiently, a priority queue, or heap, is used to hold all states that have not yet had their successors generated. The front element of the queue is guaranteed to be a state of lowest cost.

A goal state is any state that has no unremoved parts. When such a state is found, the path from it to the start state is traced. This list of states must be translated into a list of assembly actions. There are three types of action: insertion of a part or subassembly, inversion of a subassembly, and welding. Each state transition obviously implies an insertion operation, and may also require an inversion and a number of welds. The translation is straightforward; if the states joined by the transition have different orientations, an inversion

precedes the insertion; if there are welds between the inserted part or subassembly and other parts, the appropriate weld operations follow the insertion.

When the plan for the assembly is complete, a plan is computed for each declared subassembly. This continues recursively until all subassemblies have been planned, or one is encountered for which a plan cannot be found.

The computational cost of the optimizing planner is inherently related to the factorial of the number of parts in the assembly. When this method (without the use of recommended subassemblies), was applied to the A' discriminator, the search time was impractically large (it was projected to be many tens of hours of time on a high performance workstation). Adding use of recommended subassemblies to this planner was a major step, requiring substantial programming effort. The advantage of this approach is also substantial, though, because the search process for a plan for each subassembly involves far fewer parts than the whole, and a plan for the whole involves far fewer “parts” when subassemblies act in a sense as parts. Because each subassembly is planned separately, the complexity is additive, not multiplicative.

Note that when recommended subassemblies are used that the search becomes a hybrid approach, not a pure A^* search. That is, the search is not over as large a space as would be required to assure optimality. The tradeoff between computational time and optimality seems unavoidable. In practical terms, guiding the planner by suggesting reasonable subassemblies seems workable.

We use three metrics to assess the performance of the planner. The planning time, the number of states expanded, and the total number of states generated. For the A' discriminator, using the designer’s subassemblies as recommendations, these numbers are:

- Time: 50 seconds (on an SGI Indigo 2, Extreme graphics)
- Number of states expanded: 52
- Total number of states generated: 233

3.3 Use of Archimedes Illustrator

The Archimedes 2 system included an early version of an automated robot assembly simulation capability, called the “Illustrator”. The intention of the Illustrator is to be:

- independent of (that is, able to immediately adapt to) what assembly is to be simulated or what assembly plan is being used; but
- dependent on the robot workcell in which to perform the simulation.

The reason for this orientation is that there is presently no feasible way to parameterize workcell features so as to make the program able to handle any workcell. In any event, the geometry and kinematics of workcell features themselves, such as welding, gluing, plastic forming capabilities, etc, would definitely have to be user specified. The Illustrator allows one to simulate robotic assembly for any device using any plan in a simplified workcell

provided with the Illustrator, or in another workcell without all the relevant features, but realism is lost. For example, execution of an “invert” command in a workcell not containing an inverter facility or equivalent is done as a spin in place.

An overview of the operation of the Illustrator’s code is relevant at this point to help the reader understand the process. The main stages in the simulation process are:

- The CAD model parts are installed into the simulated workcell, and the parts are assembled into a replica of the Pro/ENGINEER model by referring to the assembly information generated by the Pro/ENGINEER to ACIS translator discussed above.
- The assembly plan from the Archimedes planner is read, looking for the order in which parts are referenced and the insertion direction used. The parts are moved from the assembly to a simulated “parts tray” in the order of use, and in an appropriate orientation. Parts not used in a particular plan are hidden from view.
- The assembly plan is then read a second time, and each action is processed by the core of the Illustrator, which has the form of a “finite state machine”. At every moment this program considers the workcell to be in one of a short list of states, such as assembly state, or weld state. State transitions are executed as necessary to put the workcell in the appropriate mode to execute each requested action. Each action is processed only as far as necessary. This allows the simulated behavior to adapt to each context appropriately. (For example, following a weld operation, the behavior is quite different depending on whether the next operation is another weld of the same two parts, or an unrelated assembly action.) Assembly operations may take place at a default “main” assembly point, or at any number of subassembly preparation points which can be positioned by default or in a custom layout. Fixtures may also be associated with each assembly point. Parts are picked up as needed from the parts tray, which advances forward before each pick. Needed subassemblies are picked up from their temporary assembly locations.

Generally, when applying the Illustrator to a new workcell, this state machine needs to be adapted or expanded to handle actions not previously programmed. In addition, a realistic model of the workcell must be built in CimStation. We discuss the building of the workcell model next.

3.3.1 Building the A’ Simulated Workcell

CimStation includes extensive tools for building a workcell model. Models can be imported from CAD tools and specialized robots can be described and kinematically connected into a functioning robot model. Models of common commercial robots can also be obtained from SILMA. A screen capture of the workcell with welder and shuttle is shown in Figure 3. For the A’ project the following modeling activities were required:

- A model with accurate kinematics was obtained from SILMA for the AdeptOne robot.

Figure 3: The A' Simulated Workcell

- An accurate model of the A' workcell support structure had been created by Jim Spalding of Department 2171, using IGRIP, from drawings by Bob Woods. An IGES file was exported from that environment and imported into CimStation to provide a realistic workcell appearance.
- The model for the XYZ table (which is used to position the laser weld head) was created with CimStation's local modeling capability, using simple rectangular and triangular shapes which were grouped into three stages. CimStation's Metakinematics facility was used to connect the three stages into a PPP (three prismatic joint) robot, which was then programmed within CimStation in the same manner as any other robot. A mounting bracket for the laser head was attached to the third stage, and a simple laser head model was mounted on it. The laser beam, which is flashed to illustrate the welding, was modeled suggestively as a transparent cone and affixed to

the laser head. The XYZ table's enclosure that will protect personnel from exposure to the laser beam was not modeled in CimStation, though it was fully detailed in the engineering drawings. A transparent housing could easily be added for further realism if desired, but of course a realistic rendering of the enclosure would defeat the point of letting the user "see" the weld operation.

- The shuttle which moves between the assembly area and the welder housing was modeled as a group of shapes adopted into a single object, and its motion was controlled by doing a smooth move between two taught poses. It could also have been modeled as a single rotational joint robot, but the first approach was simpler.
- The inverter was modeled in Pro/ENGINEER by Bob Woods as a part of his design process for the inverter, and was converted to CimStation using SILMA's standard Pro/E interface. Several poses had to be taught for use in producing the motion of the inverter.
- A parallel jaw gripper tool was created by taking a similar gripper model that was provided with CimStation's training materials, and scaling it to a more appropriate size. A vacuum gripper model was created by using cylinders in CimStation and using the tool generation capability to set up the mounting and actuator poses.

Each of the above individual activities was a small effort. Once one has developed some facility with CimStation's modeling capabilities, building such a workcell model is a fairly routine matter which should not be a serious barrier to doing simulations with the Illustrator. However, the availability of realistic models such as the workcell support structure model makes building the workcell model much easier (and more realistic), as the major remaining effort is then in defining the various poses of the objects.

One extra task in defining a workcell to the Illustrator is to write a short text file called a "workcell representation file" which currently just provides the CimStation file name of each gripper to be referenced and its basic specifications. (Its function is expected to be expanded later.) For example:

```
remark robot = adept
```

```
parallel_jaw file = ~/cim/projects/public/tools/cgripper.ee  
parallel_jaw grip_minimum = .15 cm  
parallel_jaw grip_maximum = 7.1 cm
```

```
bgripper file = ~/cim/projects/public/tools/smallgripper.ee  
bgripper grip_minimum = 3 cm  
bgripper grip_maximum = 7 cm
```

```
vacuum file = ~/cim/projects/public/tools/vacuum_blue.ee  
vacuum grip_minimum = 0 cm    (this is irrelevant for vacuum grippers)
```

vacuum grip_maximum = 100 cm (this is irrelevant for vacuum grippers)

3.3.2 Extending the Archimedes Illustrator's Capabilities

Until we attempted to simulate operation of the A' cell, the programming of the workcell actions in the Archimedes Illustrator had been done in a straightforward, but ad hoc, approach. When the complexities of handling all the shuttle, welder, and inverter actions needed to be added to the Illustrator for A' , it became clear that a more formal model of the simulation process was needed. The paradigm of a "finite state machine" was adopted. Under this paradigm the workcell can be in any of an arbitrary number of "states", such as "assembly" state for actions performed directly by the AdeptOne robot, or "welding" state for welding operations. Transitions between the states are programmed according to a state flow chart which can be expanded as needed[5]. The structure of the inner loop then became, in brief, (where action is a line of input to be simulated):

```
didit:=false
if not didit then
  repeat
    case cellstate of
      'assembly': doassystate(action); {action is an input line to perform}
      'weld':      doweldstate(action);
      'flip':      doflipstate(action);
      ...
    end;
  until didit
```

Here, each state is handled by a separate routine which must be able to react appropriately to every statement type. For example: the outline of the procedure that handles the "assembly" state is:

```
task doassystate(action:string);
  case verb of
    'place': {perform the action...};      didit:=true;
    'weld':  {transition to weld state...}; cellstate:='weld';
    'invert':{transition to flip state...}; cellstate:='flip';
    ...
    'quit':  {cleanup as required...}
  end;
```

This approach provides a framework for the code in which there is, as the old adage says, "a place for every thing and everything in its place". The state routines which we implemented for assembly state, inversion (or 'flip') state, and weld state are well modularized examples which should serve adequately to guide other developers in expanding the

Illustrator as needed. As the code is expanded to handle more workcells the various states and actions should become a progressively useful set of examples for new applications. Note that when a state procedure cannot actually execute a command, its job is to transition to a state that can execute it, or which will eventually get the system to a state that can execute it. The iteration through the state transitions stops when some state procedure is able to set the “didit” flag to indicate that the action was actually done.

A second extension for A' was simulation of subassembly operations, and the attendant need for multiple assembly points in the workcell. The design of the program had been quite dependent on the assumption that we would be simulating linear sequences (in which each part is placed immediately in its final position). Nevertheless, we were able to extend the program to use an arbitrary number of subassemblies. The Archimedes Assembly Plan input “language” had to be extended as part of this effort. The example in Appendix A reflects those extensions.

More on the Illustrator’s function is included in a related SAND report documenting work done in the Archimedes project as part of a Laboratory Directed Research and Development Project (LDRD [5].

3.3.3 Assembly Plan Input to the Illustrator

Once the simulated workcell is developed and the programming of the Illustrator’s states and actions are expanded as appropriate for the new cell, one can proceed to execute the simulation. The input to each simulation is:

- The CAD model of the assembly, optionally including fixtures. These models must be in CimStation format. Conversion to this format is normally done by using the CimStation interface to Pro/E, which converts individual parts to CimStation format using a few menu clicks. (We understand that a more direct and immediate input of ACIS models to CimStation is soon to be available, which will ease the tedious task of asking Pro/E to convert each part in turn.)
- An assembly plan from either the Archimedes geometric planner or the Archimedes optimizing planner, or optionally a human-written script in that format. The user can edit this plan at will during use of the Illustrator to try various sequences.

Appendix A contains an example script for the Illustrator. This particular example is a hand-modified version of a plan produced by the Archimedes geometric planner for the A' discriminator. Neither Archimedes assembly planner (the geometric planner nor the optimizing planner) has at this time the capability of outputting explicit subassembly operations, so subassembly operations were added by hand to mimic typical operations in the A' workcell for assembling the A' discriminator. The subassemblies used are those that are contained in the Pro-E model of the A' discriminator. (Note: In Appendix A, commands which do not specify which subassembly to use with an “on xxx” phrase are by default referring to subassembly number 1, or “main”).

3.4 Robot Code Generation for *A'* Discriminator Assembly

One goal of the Archimedes project is to build a library of robotic routines which can handle a wide variety of assembly tasks. The Adept V+ line-interpreter is particularly well-suited for such development. An early version of such a library was generated for Archimedes 2 (see Section 2). To facilitate the robotic assembly requirements for the *A'* project, the V+ library was enhanced to more readily adapt to a larger variety of assembly tasks. The library was also extended to allow for control of additional independent axes, and techniques new to Organization 2100 were added, including precision enhanced vision.

For the *A'* project, an assembly plan file was created by hand and processed as discussed in Section 2, because the optimizing compiler's subassembly capability was not finished when robot code was needed. In follow-on work in the Archimedes Project after the *A'* project, we plan to develop the capability to write appropriate V+ library calls directly from the planner rather than writing an intermediate language which is then compiled into V+ calls by a separate program. Thus, there will be three main forms of plan output from the Archimedes planners: V+, scripts for the Illustrator, or direct animations.

4 Development of the A' Robot Workcell

It was decided early on in the planning for the A' project that a fully functional robot workcell would be developed in order to assure that the A' project's addressing of agile manufacturing issues was adequately comprehensive and grounded in actual manufacturing issues. This focus had numerous cultural and technical impacts on the project, as will be discussed in Section 5. In this section we overview the effort that was required to build this "factory in a workcell" to assemble A' discriminators.

4.1 Workcell Support Structure

The A' robot assembly workcell has been designed around an AdeptOne robot, which is a SCARA configuration robot. The frame supporting the workcell was formed of commercially-available aluminum extrusions, supplemented as required with clear plastic and metal panels that provide personnel protection and a finished appearance. The workcell structure was designed to meet specific requirements of the A' project which were somewhat atypical of industrial robot applications. In particular, it was recognized that the cell would have to be moved several times during its life. For that reason the conventional approach which employs a ponderous base, bolted to the floor, was abandoned in favor of a kinematically-designed (three point suspension) hybrid structure that interfaces aluminum extrusions to steel members in the regions that require greatest stability. Attention was given to controlling vibration using means other than dead mass. However, for maximum utility the robot support was designed to be removable so a robot support spool can be inserted and bolted to the building floor if needed. (This option was in fact chosen in order to shorten the project's schedule, since vibration of the platform could be problematic to the high precision pick/place operations needed for the A' discriminator). To reduce weight and floorspace and to provide ample capability to view the assembly operations, a roughly octagonal floorpan was adopted.

Mechanical assembly steps can be performed anywhere within the working envelope of the AdeptOne robot, although only a portion of this workspace is actually being used for assembly operations in order to maximize precision and improve assembly speed. The full work area is approximately 300 degrees of an annular shape 31.5" in outer radius, 9" in inner radius, and 7.7" deep.

Experience with instrumentation such as upward-looking television cameras that are mounted below the work surface has underlined the importance of breaking the table into a large number of easily-handled small plates in order to allow easy access to such equipment.

It is perhaps worth mentioning that the most significant change that we would make in a future design of such a workcell would be to place less trust in an ability to align a large number of countersunk holes in baseplates with corresponding slots in an extruded aluminum structure. Several alternatives exist. For example, it may be possible to use pan head screws in oversized counterbored holes rather than countersunk screw heads (which allow no radial adjustment.) A more practical approach will be to use something resembling aircraft sheet metal assembly practice, in which a few screws (or rivets) are placed to

maintain alignment and the balance of the connections are made using holes that are drilled in situ.

An important lesson, still being learned, will come from the gathering of data which compare the dimensional stability of an integrated, movable, robot/work surface combination cell to the stability of a cell that uses the traditional concrete and iron construction. Our cell is uniquely qualified for such a study, having both capabilities.

The value of having the CAD drawing files readily accessible during assembly was apparent. In many cases it proved convenient to produce drawings that enlarged specific aspects of the design. These were used to answer questions that arose during assembly. With an average CAD system on site it is possible to do this in an interactive fashion and in a matter of minutes. Nevertheless, the wisdom of deferring some final small detail designs until the cell was partially assembled was confirmed. It was seen to be far more practical to arrange wire runs and details like panel hinges after the basic structure was in place and components could be fitted directly to it. The alternative would have been an unnecessarily large expenditure in detail drafting time.

4.2 Integral Laser Welder

The workcell contains a second robot in the form of an XYZ table with a laser weld head mounted on it. The XYZ table is a commercial product from Daedal Division of Parker Hannifin, which is controlled from the Adept controller as a PPP (i.e., three prismatic joint) robot. The laser welder is a 50 watt pulsed YAG laser spot welder from Lumonics Corporation called the "Luxtar". It is cooled by a closed-cycle refrigerator that uses water as a working fluid. A CCD television camera is mounted so it can image through the laser welder lens. This was intended for use by the Adept vision system to assure precise feature location before welding operations. However, this arrangement turned out to have such a small field of view that it was impractical to apply to part feature localization for the A' discriminator. Instead, another camera similar to the ones used to locate parts on the parts tray was added to the laser's XYZ table very near the weld head, and this camera was used for locating features.

The laser itself and its power supply are contained in a single housing that can be located several meters from the workcell, with the laser beam being conducted to the workcell by a fiber optics bundle. The bundle terminates in a lens assembly that focuses the beam. Since the depth of focus is shallow, being on the order of a millimeter, the Z motion of the XYZ table is used to do the fine focusing of the laser over a desired spot weld point. Coarse focusing is accomplished by selecting an appropriate focal length lens from an assortment of focal lengths available.

The Lumonics laser system has the option of installing beam splitters which allow a single beam to illuminate multiple spots simultaneously, or to direct the full energy of the beam through one of several software-selectable fiber bundles. Our baseline design does not use either of these features, though discussions have been held with the A' Design Team about using the split beam technique to reduce the natural migration of a pin to the side of its hole to which it is first welded.

The assembly containing the laser head is mounted outside the working radius of the robot in order to preclude any danger of collisions. For laser beam safety, the XYZ table and laser head are contained completely in a metal cabinet which has a small doorway to admit a shuttle tray. The shuttle itself is a shallow tray, equipped with a pneumatically operated clamp that has the ability to directly grasp objects. As an alternative to the gripper it also has locating pins to accurately position fixtures holding small parts. To use the laser welder, the shuttle is rotated to the robot's workspace and the robot is instructed to place the assembly or a fixture containing the assembly into the shuttle. The shuttle is then rotated into the welder housing. A panel on the shuttle occludes the doorway and actuates a redundant pair of interlock switches, allowing laser operation. The XYZ table positions the weld head to the desired location above the features to be welded. A camera is mounted on the XYZ table very near to the weld head for use in precise feature location if needed. After the welding operations are performed, the shuttle returns to the robot's workspace where the AdeptOne robot picks up the welded assembly.

4.3 Design of End Effectors

In the A' project, our goal was to design the workcell hardware so as to adapt flexibly to any "child" design in the design space. This involves automatically adapting the assembly plan to the presence/absence of certain parts, and adapting to slightly different shapes or sizes of those parts which are variable, such as the mazewheel, motors, and mounting plate. Adapting the workcell to the presence or absence of parts does not present a difficulty, but adapting to variations in part geometry can present difficulties.

There were two driving factors in the A' discriminator design which affected the design and number of grippers which were used. First, the tight clearances between mating parts dictated that precise control of the component parts must be maintained during the mating operations. Second, variations in part sizes had to be accommodated with a small number of grippers. In order to manipulate all the discriminator parts and subassemblies which were selected for robotic assembly and to sustain a high level of control, three parallel-jaw grippers were developed. The parts selected for robotic assembly varied in shape and size, with grasping widths ranging from 0.036 inches to 2 inches. These parts were easily separated into three classes based on grasping widths, so the goal we set was to design one gripper for each class of parts. Each gripper was designed with grasping features consisting of vertical and horizontal notches. These notches are used to center the parts during grasping as well as to hold them during mating operations, while placing minimal restriction on the exact shape of the parts. As will be seen in the section below on assembly operations, precision can be assured by visual checks on the already grasped parts.

Our approach worked well, as the grippers now adapt readily to the whole A' discriminator parameter space (*i.e.*, all child designs), and high precision operations are possible. In fact, the same grippers can accommodate assembly of the pattern wheel example from Archimedes 2 as well.

Each gripper in the A' cell is attached to a quick-change device so grippers can be changed automatically, and each is also equipped with a remote center of compliance (RCC)

device. The RCC devices lower lateral contact forces and avoid tool and part damage in addition to assisting with the insertion of pins in holes.

4.4 Parts Presentation and Fixtures

Having developed grippers for the component parts, it was relatively simple to design a presentation kit for all the component parts as well as assembly fixtures. The presentation pallet implemented for A' allows rapid changeover to other parts and fixtures. It consists of interchangeable pallets placed on a pegboard-like base. Each pallet presents a group of parts or constitutes a fixture. The A' parts are presented loosely mounted on pegs, with no attempt to orient them radially, as orientation is determined by vision. The parts' approximate locations on their pallet relative to registration marks are assumed to be known. For A' , four fixtures were used. Each is 2 by 3 by 0.5 inches in size. Each fixture has registration markings to accommodate Archimedes vision recognition utilities. Similar fixtures would be used for other applications of the A' cell. Each fixture is designed for use with a specific subassembly of the A' discriminator. The basic fixture designs are – at least in the case of A' parts – directly derivable from the CAD data, either by hand, or automatically by Boolean subtraction of component models (or Minkowski sums involving them) from a solid block. Only a few modifications of these simple fixtures were required, and these were mostly alignment pins which were necessitated by the very high accuracy requirements and tight clearances between mating parts.

A photograph of the A' parts and fixture presentation pallet is shown in Figure 4.

We realize that it is possible to follow a variety of approaches to achieving flexible fixturing. The fixturing technique we chose facilitated flexibility in the workcell's application, and also allowed for early demonstration of A' discriminator assembly. A quite different approach worthy of mention is that which colleagues of ours in Department 2121 at Sandia are pursuing, namely a highly adaptable and automatic form of fixturing based on modular fixtures[1]. It is our intent to follow the lead of such researchers when feasible.

4.5 Assembly Operations

The assembly operations performed in the A' cell are pickup and place operations. Adhesive dispensing needs for A' are sufficiently simple that they can be performed as an enhancement to a pick/place operation by dipping an appropriately gripped part into an adhesive before placement. (This is a planned operation which had not yet been fully exercised as of this writing.) To execute each pick and place, the procedure is as follows:

1. The end-effector is exchanged if necessary.
2. The robot-mounted camera with the shorter focal length (wider angle) is moved to the approximate location of the part on the parts tray, and the part is identified. Then, the camera with the longer focal length (narrower angle) is used to more accurately identify the part's position and orientation.

Figure 4: The A' parts and assembly fixtures on the presentation pallet.

3. The part is grasped at a planned position.
4. The gripped part is then moved over an upward-pointed camera and the precise location and orientation of the part is established with respect to the robot. This is determined using vision routines similar to those used to locate the part for pickup. (This step allows us to compensate for a variety of position and orientation errors which may have built up in presenting and grasping the part. In addition, as can be seen in Figure 4, the upward pointing camera can be quite close to the part, thus allowing for excellent vision accuracy. A variety of alternative low cost techniques have been suggested, such as [8]).
5. The target placement location is then determined in a manner similar to part pickup.
6. The part placement is performed.
7. Finally, if desired, the vision system is again used to inspect the placed part to check for correctness.

Figure 5: The A' Workcell

To provide for maximal accuracy in these assembly actions, Adept's High-Accuracy Positioning System (HPS) was used to map an area of the workcell of 13.5 by 17.0 inches. This system calibrates the robot so that the best achievable accuracy is obtained within the mapped area. Due to the clustering of the parts and fixtures on a single pallet, most assembly operations are performed within the range of the mapped area.

This careful use of vision and the HPS system, along with use of an RCC, resulted in reliable assembly operations involving even the smallest parts in the assembly, such as the 0.036 inch pin which is inserted in a minimally chamfered hole having less than 0.001 inch clearance.

4.6 Workcell Status

By the end of the A' project (March 31, 1995), the workcell was fully implemented as described above except for hand-off of parts to the welder for actual welding. An overall view of the cell is shown in Figure 5. At the end of the A' project timescale a demonstration of

the robotic assembly operations themselves was given, with excellent performance. Careful use of the vision system played a large role in this success. The welding subsystem, including the vision system, was operational except for details such as light sealing around the laser enclosure for ES&H purposes. Shuttling of parts between the robot and the welding subsystem was being integrated into the process at the end of the project, but was incomplete. We hope to tie up these loose ends as part of subsequent related projects. The incompleteness of the final demonstrations was of course frustrating to all, but it did not detract from the role the robot workcell played in focusing the *A'* project on relevant automated manufacturing issues. We expect that the workcell will be integrated into follow-on projects, most notably the AMPS (Advanced Manufacturing Prototype System) and FITS (Factory Integration Technology System) projects.

It is notable that the assembly operations which were demonstrated at the end of the project took about four minutes to perform. These operations when performed manually required about four hours, due to the microscopic alignment which is required by human assemblers.

5 Automated Assembly Process Development

The A' project is both a *process development* and a *technology development* project. That is, new or refined processes and technologies are its main product, and *actual devices* produced during the active lifetime of the project are incidental. In every case, the processes and technologies developed or enhanced by the A' project constitute an extensible *fraction* of the full processes and technologies that are required to achieve agility in the manufacturing process.

In the Automated Assembly Team of the A' project there has been somewhat more emphasis on technology development and somewhat less on process development than in other A' teams due to the relative youth of the assembly planning technology, and the Archimedes project's focus on developing that technology. However, there was a significant focus on process development in this team, as will be discussed here.

5.1 Technologies for Design Feedback Regarding Assembly

The core Archimedes assembly sequencing technology, which was advanced significantly during the A' project, is of great potential in providing easily visualized feedback to designers regarding assemblability of complex designs, and in developing optimal assembly plans (where "optimal" can be a customer-defined term). The Archimedes project is already testing its planners with real industrial CAD data from potential industrial partners, with significant success. We look forward to promulgating and expanding this important tool in the coming months and years.

In some cases the technology to support process improvement was produced by the Automated Assembly Team, but there was little time left to apply it within the time frame of the A' project. For example, realistic robot illustration of the A' workcell operating on the A' discriminator itself was not fully done because the subassembly extensions to the optimizing planner were just completed at the end of the A' project, and the capability to output plans with explicit subassembly commands was not yet added by the end of the project. (Simulations of the A' cell doing simpler assembly sequences *were* performed, and illustration of subassembly use was tested with hand written examples.) We believe that the ability to efficiently produce realistic simulations very early in future projects will benefit such manufacturing projects at Sandia. Some such projects are in the planning stage now.

5.2 Collaboration with Design, Manufacturing and Other A' Teams

A very effective process improvement to which the Automated Assembly Team contributed was the ongoing collaboration that was arranged between all the teams that were directly concerned with assembly of the A' discriminator devices. Early in the project joint meetings were held including representatives of the A' Design Team and A' Automated Assembly Team. These meetings gave indication of the interesting possibilities of such collaboration. For example, the assembly team was able to suggest more direct ways of specifying alignment between critical parts, because they were aware of the capabilities that in-cell vision systems

would provide. Also, certain design-for-assembly issues were easily handled at that stage. For example, it was very easy at that early stage to change the type of screw heads to a type more amenable to robotic assembly. The effectiveness and efficiency of this early inter-team collaboration was apparent, and we now consider such collaboration to be essential to agile manufacturing projects.

Subsequent to these early specifically called meetings, representatives of all the A' teams routinely attended periodic meetings called by the Quality Assurance Team. Though not planned specifically for cross-team educational purposes, these meetings provided opportunities for the assembly team to educate other team representatives about robotic assembly issues. For example, discussions of expected reliability of robotic assembly operations naturally gave an opportunity for the assembly team to educate the other team members on the function of robotic RCC (Remote Center of Compliance) insertion hardware. Again, having seen the effectiveness of these ongoing discussions, we view them as essential to agile manufacturing.

Later in the A' project, when the team began to address specific details such as insertion forces required by certain matings, ongoing collaboration between the Design, Manufacturing and Assembly teams became routine and crucial to an effective process. For example, insertion and welding of the pin in the pin wheel is a critical operation. Non-robotic assembly methods involved use of very tight clearances (in fact, interference fits) as an attempt to assure precise positioning. But, the robot is not able to directly exert forces high enough to execute such insertions. Considerable inter-team discussion eventually led to a new method involving use of a precise fixture that assured verticality of the pin in its final position and allowed opening of the clearances to a level that could be executed by the robot. And, related ideas arose, such as the simple idea of doing the first spot weld at a position which would allow pin drift due to the weld to be compensated for in the final alignment procedure. (Such adjustment is only possible in the direction of rotation, not the radial direction.) More routine matters, such as decisions to chamfer certain holes or pins to assure reliability of assembly operations were also handled in these collaborations.

As mentioned above, the team chose the use of simple custom fixtures as the optimum for our particular situation. Such an approach would not have been feasible were it not for the high degree of cooperation which was developed between the personnel in the manufacturing shops and the automated assembly team. Such cross-functional teamwork now seems to us to be an inherent part of an agile manufacturing process.

6 Summary

The Automated Assembly Team's participation in the A' project has been beneficial in two ways:

- The Automated Assembly Team was able to help the A' project produce a superior agile manufacturing process, and a superior product, with design-for-assembly issues addressed at various stages of the project.
- The Archimedes project members and the members of the A' robot team were able to hone their technologies and personal knowledge and skills on a challenging assignment, with the results that
 - more capable, realistically based software was developed by the Archimedes project, and
 - considerable experience was gained in the realities of designing, fabricating, assembling, integrating, and programming a flexible robotic assembly workcell.

We hope that other projects working on agile manufacturing will be able to benefit from the automated assembly technologies developed for A' , and from the experience gained during this effort.

Acknowledgements and Trademarks

We wish to thank Irl Vance of Mechanical Processing Department 2481 for his close work with the A' Automated Assembly Team, which resulted in the fabrication of parts presentation and fixturing hardware in much shorter time than usual.

Pro/DEVELOP and Pro/ENGINEER are trademarks of Parametric Technology Corporation.

CimStation and SIL are trademarks of SILMA, Incorporated.

ACIS is a trademark of Spatial Technology Incorporated.

GL is a trademark of Silicon Graphics, Incorporated.

V+ is a trademark of Adept Technology, Incorporated. AdeptVision AGS is a product of Adept Technology, Incorporated.

A Example Assembly Plan Input to Archimedes Illustrator

```
subassembly 1 is main
subassembly 2 is r37955
subassembly 3 is r37854
subassembly 4 is r37910
fixture 2 is block_02
(place r37814_01 on main by 0, 1, 0 width 1.718 using parallel_jaw)
(place r37783_06 on main by 0, 1, 0 width 0.245)
(place r37768_04 on main by 0, 1, 0 width 0.21385)
(place r37770_05 on main by 0, 1, 0 width 0.21385)
(place ms16555_02 on main by 0, 1, 0 width 0.0938)
(place ms16555_03 on main by 0, 1, 0 width 0.0938)
(place r35295_17 on r37955 by 0, 1, 0 width 1.46386)
(place r37787_18 on r37955 by 0, 1, 0 width 0.290948)
(place r37769_30 on r37854 by 0, 1, 0 width 0.666804)
(place r37853_28 on r37854 by 0, 1, 0 width 1.03816)
(place r37844_29 on r37854 by 0, 1, 0 width 0.0497856)
(place r37782_13 on r37910 by 0, 1, 0 width 0.554224)
(place r37781_12 on r37910 by 0, 1, 0 width 0.670403)
(place r37909_09 on r37910 by 0, 1, 0 width 2.06071)
(place r37813_10 on r37910 by -0.465841, 0, -0.884868 width 0.01)
(place r37843_11 on r37910 by 0, 1, 0 width 0.0445355)
invert r37955 using vacuum
place r37955 by 0,-1, 0 width 1.46386
(place ms51957_24 by 0, 1, 0 width 0.161 using parallel_jaw)
(place ms51957_25 by 0, 1, 0 width 0.161)
(place r35295_15 by 0,-1, 0 width 1.43258)
(place ms51957_23 by 0, 1, 0 width 0.161)
(place ms51957_22 by 0, 1, 0 width 0.161)
(place switch_37 by 0, 1, 0 width 0.554337)
(place panhd0_40 by 0, 1, 0 width 0.11487)
(place panhd0_39 by 0, 1, 0 width 0.11487)
(place r37782_31 by 0, 1, 0 width 0.480846)
(place bear125_19 by 0, 1, 0 width 0.296)
(place bear125_35 by 0, 1, 0 width 0.296)
(place rr510012_34 by 0, 1, 0 width 0.194185)
(place r37787_16 by 0, 1, 0 width 0.291475)
place r37854 by 0, 1, 0 width 1.04
(place bear125_08 by 0, 1, 0 width 0.296)
(place r37836_07 by 0, 0,-1 width 0.475908 coil spring)
(place r37860_14 by 0, 1, 0 width 0.605644 ratchet pin)
(place rr51336_36 by -0.0995037, 0, -0.995037 width 0.01)
(place bear125_32 by 0, 1, 0 width 0.296)
(place rr510012_33 by 0, 1, 0 width 0.194185)
place r37910 by 0, 1, 0 width 2.06 ?
```

References

- [1] R. C. Brost and K. Y. Goldberg. A complete algorithm for synthesizing modular fixtures for polygonal parts. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, pages 535–542, 1994.
- [2] K. V. Diegert, R. G. Easterling, M. R. Ashby, G. L. Benavides, C. Forsythe, R. E. Jones, D. B. Longcope, and S. W. Parratt. Achieving agility through parameter space qualification. In *Proc. of the Agile Manuf. Enterprise Forum*, 1995.
- [3] C. Forsythe, M. R. Ashby, G. L. Benavidez, K. V. Diegert, R. E. Jones, D. B. Longcope, and S. W. Parratt. A process for the agile product realization of electro-mechanical devices. Technical report, Sandia National Laboratories, March 20 1995. (to be published).
- [4] L. S. Homem de Mello and S. Lee, editors. *Computer-Aided Mechanical Assembly Planning*. Kluwer Academic Publishers, 1991.
- [5] S. G. Kaufman, A. L. Ames, T. L. Calton, R. E. Jones, C. A. Laguna, and R. H. Wilson. Automated planning and programming of assembly of fully 3-d mechanisms. SNL report in preparation, 1995.
- [6] N. J. Nilsson. *Principles of Artificial Intelligence*. Springer-Verlag, 1980.
- [7] D. R. Strip and A. A. Maciejewski. Archimedes: An experiment in automating mechanical assembly. In *Proc. of the 11th ASME Intl. Conf. on Assembly Automation*, 1990.
- [8] A. S. Wallack, J. F. Canny, and D. Manocha. Object localization using crossbeam sensing. In *1993 IEEE International Conference on Robotics and Automation*, volume 1, pages 692–699. IEEE Robotics and Automation Society, May 1983.
- [9] R. H. Wilson. *On Geometric Assembly Planning*. PhD thesis, Stanford Univ., March 1992. Stanford Technical Report STAN-CS-92-1416.
- [10] R. H. Wilson, A. L. Ames, T. L. Calton, R. E. Jones, S. G. Kaufman, and C. A. Laguna. Lessons learned from a second generation assembly planning system. In *1995 IEEE International Symposium on Assembly and Task Planning*, 1995.
- [11] R. H. Wilson and J.-C. Latombe. Geometric reasoning about mechanical assembly. *Artificial Intelligence*, 71(2):371–396, 1994.

Distribution:

MS 9018 Central Technical Files, 8253-2 (1)
MS 0899 Technical Library, 13414 (7)
MS 0619 Print Media, 12615 (1)
MS 0100 Document Processing for DOE/OSTI, 7613-2 (10)
MS 0951 Rondall E. Jones, 2121 (30)

Single copies to:

MS 0322 Patrick J. Eicker, 2100
MS 0439 Don Longcope, 1434
MS 0960 Jimmie Searcy 2400
MS 0561 Tommy Simpson, 2481
MS 0561 Irl Vance, 2481-2
MS 0641 Ken Eras, 2643
MS 0660 Margaret Olson, 2122
MS 0660 Arlo Ames, 2122
MS 0660 Steffen Parratt, 2122
MS 0829 Kathleen Diegert, 12323
MS 0829 Chris Forsythe, 12323
MS 0949 Ray Harrigan, 2102
MS 0949 Perry Molley, 2111
MS 0951 David Strip, 2121
MS 0951 Randy C. Brost, 2121
MS 0951 Terri L. Calton, 2121
MS 0951 Cathy A. Laguna, 2121
MS 0951 Stephen G. Kaufman, 2121
MS 0951 Randall H. Wilson, 2121
MS 0958 Gilbert Benavides, 2484
MS 1006 Pablo Garcia, 2171
MS 1006 Robert O. Woods, 2171
MS 1006 Mike Saavedra, 2171
MS 1006 J. Pete Rebeil, 2171
MS 1007 Alfred Jones, 2172
MS 1176 Robert Palmquist, 2151
MS 1177 Dan Horschel, 2161
MS 1177 Michael Griesmeyer, 2161