Proceedings of the 1996 IEEE
International Conference on Robotics and Automation
Minneapolis, Minnesota - April 1996

# Design of an Agile Manufacturing Workcell for Light Mechanical Applications

Roger D. Quinn, Greg C. Causey
Department of Mechanical and Aerospace Engineering
Frank L. Merat, David M. Sargent, Nicholas A. Barendt
Wyatt S. Newman, Virgilio B. Velasco Jr.
Department of Electrical Engineering and Applied Physics
Andy Podgurski, Ju-yeon Jo
Leon S. Sterling, Yoohwan Kim
Department of Computer Engineering and Science

Case Western Reserve University (CWRU)
Cleveland Ohio, 44106

## Abstract

This paper introduces a design for agile manufacturing workcells intended for light mechanical assembly of products made from similar components (i.e. parts families). We define agile manufacturing as the ability to accomplish rapid changeover from the assembly of one product to the assembly of another product. Rapid hardware changeover is made possible through the use of robots, flexible part feeders, modular grippers and modular assembly hardware. The flexible feeders rely on belt feeding and binary computer vision for pose estimation. This has a distinct advantage over non-flexible feeding schemes such as bowl feeders which require considerable adjustment to changeover from one part to another. Rapid software changeover is being facilitated by the use of a real-time, object-oriented software environment, modular software, graphical simulations for off-line software development, and an innovative dual VMEbus controller architecture. These agile features permit new products to be introduced with minimal downtime and system reconfiguration.

## 1. Introduction

### 1.1 What is Agile Manufacturing ?

Agile manufacturing is a term that has seen increased use in industry over the past several years. The definition of "agile", however, is not clear, nor is it consistent: "Agility: The measure of a manufacturer's ability to react to sudden, unpredictable change in customer demand for its products and services and make a profit"[1]. "Today factories are coming on line that are agile at tailoring goods to a customers requirements, without halting production..."[2]. "Agile manufacturing assimilates the full range of flexible production technologies, along with the lessons learned from total quality management, 'just-in-time' production and 'lean' production"[3]. The only common thread among the various definitions is the ability to manufacture a variety of similar products based on what may be rapidly changing customer needs. In the past, production was geared toward high-volume production of a single product. In today's market, however, the emphasis is moving toward small lot sizes from an ever-changing, customer-driven product line.

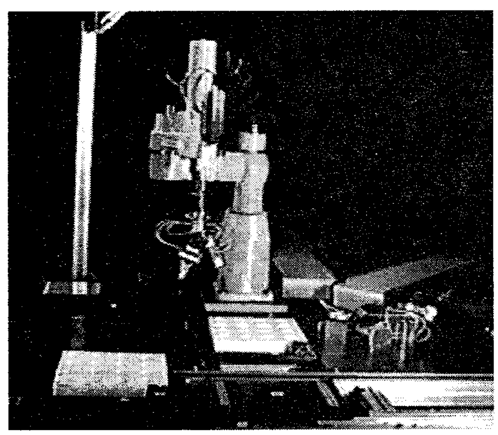

**Figure 1: Agile Workcell**

A definition of "agile" manufacturing has been adopted which applies to light mechanical assembly of products: *Agile manufacturing is the ability to accomplish rapid changeover between the manufacture of different assemblies utilizing essentially the same workcell.* Rapid changeover (measured in hours), further, is defined as the ability to move from the assembly of one product to the assembly of another product with a minimum of change in tooling and software. Rapid changeover enables the production of small lot sizes, allowing for 'just-in-time' production. A central theme of our definition of "agile" manufacturing is the ability to *rapidly introduce* (measured in weeks) new assemblies and components into the system.

In this system, rapid changeover is accomplished through the use of reusable software, quick change grippers for the robotic manipulators, modular work tables, and parts feeders which are flexible enough to handle several types of parts without needing mechanical adjustment. These feeders use vision, in place of hard fixturing, to determine the position and orientation of parts. Generic, reusable vision routines permit new parts to be added to the system with a minimum of effort.

A testbed implementation of an agile manufacturing workcell has been developed (Figure 1). This includes mechanical manipulators, flexible part feeders, a vision system (cameras, frame grabber, and a library of image processing routines), as well as a limited number of dedicated sensors and actuators needed to complete a given assembly. The central feature of such a workcell is a controller capable of controlling each of the aforementioned components.

### 1.2 Relevance of CWRU Work

Several companies have implemented what may be considered "agile" manufacturing. Motorola has developed an automated factory with the ability to produce physically different pagers on the same production line[4]. At Panasonic, a combination of flexible manufacturing and just-in-time processing is being used to manufacture bicycles from combinations of a group of core parts[5].

Against the backdrop of such work, the CWRU workcell is innovative in several ways. The use of vision-guided, flexible parts feeders is one example. Another is the development of software design patterns for agile manufacturing. The over-arching design philosophy of quick-changeover, however, is what makes this workcell particularly novel. The CWRU workcell has been designed to be a versatile production facility, amenable to a wide range of applications and an enabling technology for factory wide agile manufacturing.

## 2. Workcell Hardware

The agile workcell developed at CWRU consists of a Bosch flexible automation system, multiple Adept SCARA robots, as many as four flexible parts feeders per robot, and an Adept MV controller. An important feature of the workcell is the central conveyor system, which was implemented using standard Bosch hardware. It is responsible for transferring partially completed assemblies between the robots and for carrying finished units to an unloading robot The robots are mounted on pedestals near the conveyor system. Pallets with specialized parts fixtures are used to carry assemblies throughout the system, after which the finished assemblies are removed from the pallet by the unloading robot. Finally, a safety cage encloses the entire workcell, serving to protect the operator as well as providing a structure for mounting overhead cameras.

### 2.1 Conveyor System

The conveyor system used in the CWRU workcell is a model T2 manufactured by Bosch. Pallets are circulated on two main conveyor lines. These lines are parallel to each other and operate in opposite directions. Pallets are transferred between these two sections by means of Lift Transfer Units (LTU's). These allow for the circulation of pallets around the conveyor system and the capability to re-order, the pallets.

Each of the pallets in the system is given a unique identification number, allowing the system to track and direct its progress. Stops are mounted at critical points on the conveyor to control the flow of the pallets.

An innovative use of this conveyor system is the use of short "spur lines". A spur (Figure 2) is simply an extension of the conveyor, perpendicular to the main line (analogous to a railroad spur). This allows the flow of the main conveyor line to be maintained while a robot performs an assembly at the spur. Pallets entering a spur are registered in the robot's world coordinate frame by an arm-mounted camera, allowing the robot to place or remove parts on the pallet and avoiding the expense of mechanical registration.

### 2.2 Assembly Stations

Several assembly station layouts were analyzed in choosing the final layout. After evaluating several features of each layout, including: placement of the robots relative to the conveyor, impact of feeder placement relative to the robot work envelope, and the robot motions necessary for a generic assembly, it was determined that the layout in Figure 2 would best suit the needs of the workcell.

Each assembly robot is surrounded by two modular, removable work tables and two fixed feeding tables (Figure 2). The modular tables are easily exchangeable, allowing for specialized assembly hardware to be placed within the robot's work envelope. The modular tables contain pneumatic actuators and electrical sensors which can be connected quickly, allowing the rapid change of any specialized tooling required for a given assembly. By designing these tables to be modular and easily exchanged, different assembly hardware can be quickly accommodated. To achieve rapid changeover, the modular work tables are registered in the robot's world coordinate system in the same manner as the pallets (i.e. using an arm-mounted camera). The feeding tables are fixed, and the horizontal, parts-feeding conveyors are mounted to them.

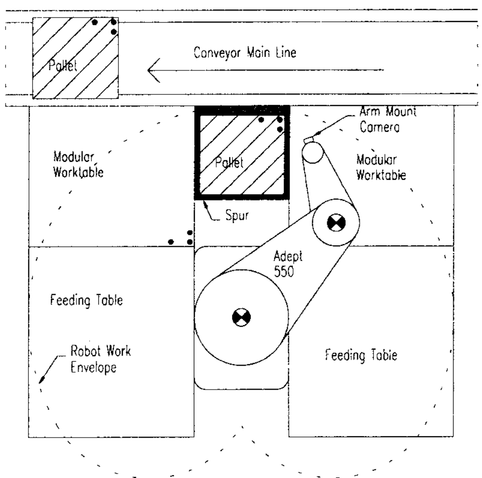


**Figure 2: Workstation Layout**

One drawback of the conveyor/spur system, as outlined above, is the time required to exchange a full pallet for an empty one. During this time (approximately 15 seconds) the robot would conceivably be inactive. A simple solution to this problem is a mini-warehouse: a fixture is located on the exchangeable portion of the work table to hold a few completed assemblies. During a pallet swap, the robot can continue the assembly operation working while the incoming pallet arrives, placing the completed assemblies in the mini-warehouse. After the incoming pallet is transferred to the spur, the vision system registers the pallet. The robot places the current assembly (still in its gripper) on the pallet and then proceeds to move the completed assemblies from the mini-warehouse to the pallet.

## 2.3 Flexible Parts Feeders

Each feeder consists of three conveyors (Figure 3). The first conveyor is inclined and lifts parts from a bulk hopper. The second conveyor is horizontal, with a translucent belt. It transports the parts to the robot, presenting them at an underlit section near the robot. The third conveyor returns unused or unfavorably oriented parts to the bulk hopper.

Proper functioning of the feeders depends on the parts being lifted from the bulk hopper in a quasi-singulated manner. Many factors influence the effectiveness of the inclined conveyor: the angle of the conveyor with respect to the horizontal, the belt properties (e.g. coefficient of friction), the type of belt (cleated, magnetic, vacuum), and the linear speed of the belt, for example.

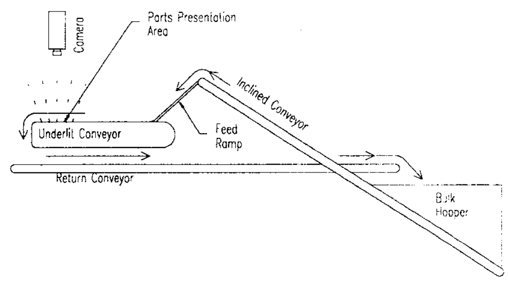


**Figure 3: Flexible Feeding System Schematic**

When different parts are to be fed, the bulk hopper is emptied and filled with the new parts. If the parts are of a similar geometry, no changes to the feeding system are typically needed. Some parts, such as circular or cylindrical ones (i.e. ones that would roll back down the incline) may need a different belt surface, such as a cleated one, or a different angle of inclination.

Overhead cameras are used to locate parts on the horizontal conveyors. An array of compact fluorescent lights is installed within each of the horizontal conveyors. These lights together with a translucent conveyor belt provide an underlit area in which parts can be presented to the vision system. Using binary vision tools (currently provided by an Adept vision system) parts on the feeder belts are examined. First, the vision system looks to see if a part is graspable (i.e. the part is in a recognized, stable pose (position and orientation) and enough clearance exists between the part and it's neighbors to grasp it with a gripper). Second, the pose of the part in the robot's world coordinates is determined. This pose, and the motions associated with acquiring the part, are checked to make sure that they are within the work envelope of the robot.

## 2.4 Vision System

One essential function of the vision system is to determine the pose of components for flexible parts feeding. Pose estimation is performed using built-in functions of the AdeptVision software, and must be fast enough not to interfere with the assembly cycle-time. A secondary function of the vision system is to register pallets and modular work tables to a robot's world coordinate system, avoiding the need for alignment hardware. Still another use may be error recovery, wherein the cameras can be used to inspect critical points in the system, or in-process assemblies.

The vision system uses a number of standard CCD cameras, mounted either above the flexible parts feeders or on the robot arms. Since the number of camera inputs to the AdeptVision system is limited to four, a low-cost, custom video multiplexer was developed, utilizing a monolithic video-switcher integrated circuit. This allows up to four cameras to be attached to each video input on the video hardware.

In keeping with the quick-changeover philosophy, the vision routines are designed to be reusable; that is a given routine may be used to locate several different but similar parts (i.e. similar asymmetries, topology, etc.). This approach has many advantages, including minimizing the number of software routines. In addition, this reusability allows for software modularity and "agility[6]." For example, by parameterizing the characteristics that a routine searches for, it can be applied to parts that have a similar profile but are of a different size. This means that parts with similar geometries to those in the parts library can be added to the system by simply modifying the inspection procedures that call these lower-level, reusable routines.

### 2.5 Introduction of New Parts

Adding a new part to the system involves a few well defined tasks. A vision routine which determines the pose of the part is developed, utilizing the library of reusable vision routines. If the new part has characteristics that appear nowhere else in the parts library, new routines may need to be added to the software library. Also, if the part has not been designed for use on the generic parts feeders (e.g. it has no stable poses, like a cylinder), the feeders may require a belt change or a change in the angle of inclination. A gripper must also be designed to manipulate the new part. In order to minimize the specialized hardware and avoid tool changes during assembly, the gripper design should be performed concurrently with the gripper designs for other parts to be assembled at a given robot. For instance, if a given operation requires both an A widget and a B widget to be assembled at the first robot, the gripper designer should take this into account.

In general, it is best to design the parts and the associated hardware concurrently. This will allow maximum reuse of software, minimal change to the flexible feeding setup and the design of a robust assembly sequence which will enhance unattended operation. This approach is known as Design for Manufacturing and Assembly[7], or simply DFMA.

## 3. Computer Hardware/Controller Design

The current software has been developed entirely in the V+[8] programming language and operating system, on Adept's MV controller. For most industrial applications, this programming environment would be sufficient; however, it lacks the power and flexibility needed to support rapid software development and changeover. This is largely because V+ lacks features which are standard in other languages and operating systems, such as user-defined functions, standard data structures and shell script execution.

To circumvent these limitations, a more extensive controller interface design is under development. It will allow the system to support C and C++, and provide a friendlier and more flexible user interface. In addition, it will allow the use of a real-time operating system, thus simplifying software development and improving performance.

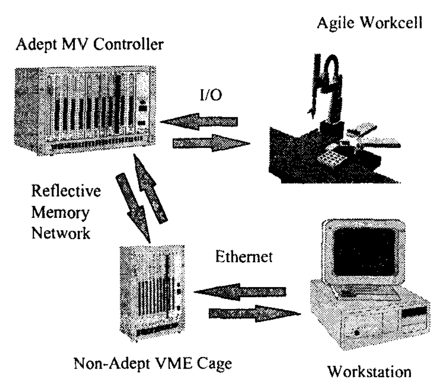


**Figure 4: System Architecture**

In this design, the system's capabilities are expanded by using a second VMEbus in addition to the MV controller VMEbus (Figure 4). This second VMEbus houses I/O boards and dedicated single-board computers (SBC's), on which a real-time operating system executes. C and C++ programs running on the SBC's are responsible for all high-level control and robot motions (e.g. conveyor control, pneumatic operations, specifying robot destinations), while the MV controller is used exclusively for low-level robot motions (e.g. servo control and trajectory generation) and some machine vision routines. In later implementations, a vision processing board can also be used on the second VMEbus, thereby augmenting the AdeptVision system.

The two buses are connected by a reflective memory network. This consists of two memory cards, one on each bus, which can be connected by either a cable or a fiber optic link. Changes made to memory on one board are automatically reflected on the other, thus allowing commands and data to be transmitted between the two buses[9]. The SBC's can thus place robot and vision commands on the reflective memory network. These are read by a set of command servers running on the MV controller. The servers execute the commands and, where applicable, return the results via the same network.

## 4. Workcell Software

Software is another key to the flexibility of an agile manufacturing workcell; however, this flexibility does not come without careful design. Although software is inherently easier to change than hardware, the structure of a software system can degrade after repeated modification, leading to poor reliability and increased maintenance costs. In designing the workcell control software, we have employed software engineering methods and tools that support the principle of design for change. In particular, our latest design is object-oriented (OO), that is, it is based upon identifying the objects of the system, which are those entities having a state and a behavior. Physical devices, abstract data structures, and entire subsystems are modeled as objects that provide a well-defined set of services whose implementation is encapsulated and hidden. Emphasis on OO software design reduces the amount of effort required to introduce new products into the workcell through reusable code. The ability to rapidly introduce new products into the workcell is crucial to an agile system.

Object types or classes are defined using the C++ class construct, wherein services correspond to member-function calls. New classes are derived from existing ones by adding services or by overriding the implementations of existing services. Object orientation facilitates maintenance because the implementation of a class can be changed without affecting client code, which uses the class's services, and because a derived class can be used wherever its parent class can be.

In addition to satisfying the requirements of our particular manufacturing application, we wish to specify software design components that might prove useful in a variety of agile manufacturing applications. Hence, we have sought to identify design patterns for agile manufacturing. A design pattern is a group of communicating objects or classes which together represent a reusable design element that is applicable, after some specialization, to a variety of systems[10]. So far, we have identified design patterns for such activities as overall system control, communication between system components, provision of parts and assemblies, scheduling of system tasks, and error handling.

### 4.1 Operating System

The initial versions of the workcell control software were implemented with the V+ operating system and programming language provided with the Adept MV controller. Although V+ provides adequate facilities for many robotic applications, we determined that a more advanced operating system and programming language would better support our software design philosophy and the goals of agile manufacturing. In general, workcell control involves the management of a number of concurrent tasks with real-time constraints. Hence, a real-time operating system (RTOS) with sufficient and reliable facilities for task scheduling, communication, and synchronization is desirable.

### 4.2 Software Architecture

The workcell control software is designed as a hierarchy of servers. At the highest level, the workcell controller services requests from the human operator for crates of finished assemblies. In doing so, it communicates with subordinate servers. It makes requests to the pallet server to move pallets along the conveyor between spurs. The pallet server tracks the movement of each pallet, but the workcell controller is responsible for knowing pallet contents. The workcell controller makes requests to assembly servers, which are associated with robots, to fill pallets with partial or complete assemblies. To satisfy these requests, the assembly servers must communicate with subordinate parts servers as well as with the robots and special purpose hardware. The parts servers in turn must communicate with parts feeders and with the vision system. In general, servers are designed with as few assumptions about the overall workcell structure as possible, so that they are not sensitive to changes in that structure. Where appropriate, servers operate concurrently; for example, while a robot adds a part to an assembly, the server for that part attempts to locate another part in anticipation of the robot's next request.

Error handling is also hierarchical. If a server encounters an error condition, it first tries to resolve it locally, e.g., by making additional requests to subordinate servers. If this fails, the server indicates to its client that it was unable to provide the requested service. The client then tries to resolve this error condition. As a concrete example, consider a part server that is unable to locate a part in its vision window. It will repeatedly advance the feeder and take pictures. If no part is found within a certain number of repetitions, the parts feeder will report failure to the assembly server. If there are no redundant parts feeders which it can invoke, the assembly server will signal failure to the workcell controller. In the absence of redundant servers for the assembly in question, the controller will inform the operator of a problem requiring human intervention. Robust, fault-tolerant software is necessary for unattended-operation of the workcell.

### 4.3 Workcell Simulation

As the software development progressed concurrently with the construction of the hardware system, it became evident that an emulation of the expected hardware system would be extremely useful. The decision was made to begin development of a comprehensive simulation that would permit the workcell control code to be developed and tested without using the actual hardware. This allows the control software to be written and debugged without halting the production of a functioning workcell. It also proved useful to display the hardware response graphically (Figure 5), especially for investigating various workcell layouts.

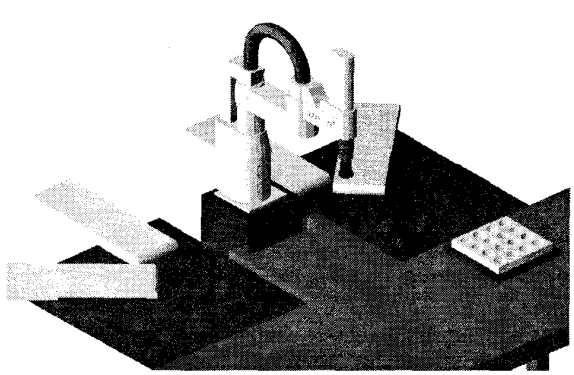

**Figure 5: TELEGRIP Simulation**

The conveyor system has been completely simulated. Detailed simulation of the robots and vision system is under development. The simulation code mimics the inputs and outputs of the workcell, allowing for transparent use of the simulation. In other words, the code which is used to simulate the control of the workcell is the **same** code running on the **same** processor boards used to control the physical plant. This is a powerful tool for software design in that there are no inconsistencies between the simulation control code and the actual control code. This eliminates possible porting problems in moving from simulation to the actual control platform.

## 5. Conclusions

This research successfully validates the critical issues for the design of an agile manufacturing workcell. Flexible parts feeders, machine vision, modular hardware, an extensive controller interface, on-line error correction, graphical simulations and modular software are all essential elements of an extensive implementation.

In continuing work, the system is being expanded to include modular vision routines, the use of a real-time operating system, object-oriented programming, and extensive error detection and recovery. Product design for manufacturing and assembly will also play a key role in facilitating feeding, assembly, and pose estimation.

## 6. Acknowledgments

This work was supported by the Cleveland Advanced Manufacturing Program (CAMP) through the Center of Automation and Intelligent Systems Research (CAISR) and the Case School of Engineering.

---

[1] P.M. Noaker. The search for agile manufacturing. *Manufacturing Engineering*, 30:28-42, 1993.

[2] R. Comerford. The flexible factory: Case studies. *IEEE Spectrum*, 30:28-42, 1993.

[3] Steven L. Goldman and Roger N. Nagel. Management, technology and agility: the emergence of a new era in manufacturing. *International Journal of Technology Management*, 8(1/2), 1993.

[4] R. Strobel and A. Johnson. The flexible factory: Case studies. *IEEE Spectrum*, 30:28-42, 1993.

[5] T.E. Bell. The flexible factory: Case studies. *IEEE Spectrum*, 30:28-42, 1993.

[6] David Michael Sargent. A framework for computer vision in agile manufacturing. Master's Thesis, Case Western Reserve University, 1996.

[7] Boothroyd Dewhurst, Inc. *Product Design for Assembly*, March 1991

[8] Adept Technology Inc., *V+ Language Reference Guide, Version 11.0*, June 1993

[9] S. May. Using reflective memory to build highly interactive real-time multiprocessing systems. *VMEbus Systems*, 9(3), 1992.

[10] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns-Elements of Reusable Object-Oriented Software*. Addison Wesley, Reading, MA, 1994.