Proceedings of the 1997 IEEE
International Conference on Robotics and Automation
Albuquerque, New Mexico - April 1997

Advances in Agile Manufacturing

Frank L. Merat[*], Nicholas A. Barendt[*]
Roger D. Quinn[†], Greg C. Causey[†]
Wyatt S. Newman[*], Virgilio B. Velasco, Jr.[*]
Andy Podgurski[‡], Yoohwan Kim[‡]
Gultekin Ozsoyoglu[‡], Ju-Yeon Jo[‡]

Center for Automation and Intelligent Systems Research (CAISR)
Case Western Reserve University (CWRU)
Cleveland, Ohio 44106

Abstract

An agile workcell has been developed for light mechanical assembly in collaboration with industrial sponsors. The workcell includes multiple Adept robots, a Bosch conveyor system, multiple flexible parts feeders at each robot's workstation, CCD cameras for parts feeding and hardware registration, and a dual VMEbus control system. Our flexible parts feeder design uses multiple conveyors to singulate the parts and machine vision to locate them. Specialized hardware is encapsulated on modular grippers and modular worktables which can be quickly interchanged for assembly of different products. Object-oriented software (C++) running under VxWorks, a real-time operating system, is used for workcell control. An agile software architecture was developed for rapid introduction of new assemblies through code re-use. A simulation of the workcell was developed so that controller software could be written and tested off-line, enabling the rapid introduction of new products.

1. Introduction

The agile-manufacturing project at Case Western Reserve University, publicly unveiled in 1996 at this same conference[1,2], is a multidisciplinary project combining the efforts of students and faculty from three departments (Electrical Engineering, Mechanical Engineering, and Computer Engineering and Science), as well as active engineering collaboration from local industry. The purpose is to construct a system for light mechanical assembly that is capable of rapid product changeover as well as rapid introduction of new product designs[1]. Our objectives are to develop specific techniques, to identify and present relevant lessons, and to establish general principles and philosophies for the design of agile manufacturing systems[3]. This knowledge is being gained and experimentally validated through the construction and operation of a sophisticated, yet industrially-relevant, testbed.

Our experimental testbed includes commercial components integrated with novel designs. Commercial components include: three Adept robots; an Adept MV controller with AdeptVision; 8 CCD cameras; and a Bosch flexible material-handling system. Custom additions include: flexible part-feeder designs, creative gripper designs, and a flexible supervisory controller. Less obvious but more important are the lessons learned from the experience, including: modular and generic gripper, feeder and tooling design approaches that enable rapid changeover; lessons for simultaneous product/process design with extensions of "design for manufacturability" particular to agile manufacturing; and philosophy and techniques supporting reusability, rapid modification and maintainability of software.

This paper surveys the progress achieved over the last year, extending the results presented in Quinn et al[1]. More extensive detail in focus areas can be found in an upcoming journal paper[4], parallel submissions to this conference[5,6,7,8], and technical reports[9,12].

2. Mechanical Design Strategies for Agile Manufacturing

As introduced in Quinn et al.[1], our system includes several critical elements which enable agility. Chief among the physical aspects are: vision-based flexible parts-feeder systems; the introduction of "spurs" into

[*] Department of Electrical Engineering and Applied Physics
[†] Department of Mechanical and Aerospace Engineering
[‡] Department of Computer Engineering and Science

the conveyor system; the use of modular worktables supporting specialty fixturing and tooling; and multi-purpose, modular end-of-arm tooling (Figure 1). Further progress has been made in each of these areas.

2.1 Flexible Parts Feeders

While flexibility encompasses every part of the workcell design, including hardware and control software, the ability to feed parts with a wide variety of sizes and shapes is crucial. Conventional feeding methods, such as vibratory bowl feeders, are not practical for flexible workcells because of their specialized nature. When a new or different assembly is produced, the parts relating to the new assembly need to be fed without downtime for designing, tuning, and installing a new feeding system. Several flexible parts feeders are currently being marketed. A major drawback to the current designs has been their limited capacity, both in terms of part size and hopper volume. In contrast, our design, introduced in Quinn et al.[1], allows the use of larger parts (roughly softball-sized) and large hoppers.

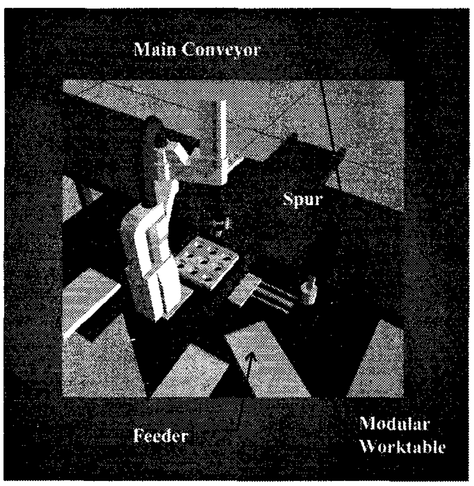

Figure 1: Assembly Station Layout (from Simulation)

Our flexible feeder design[7] consists of three conveyors working in concert. The first conveyor is inclined and is used to lift parts from a bulk hopper in a quasi-singulated manner. By adjusting the angle of inclination, speed of the belt, and belt material, the throughput of the inclined conveyor can be altered. Parts fall from the end of the inclined conveyor onto the second conveyor. This horizontal conveyor terminates within the reach of the robot in an underlit window which provides backlighting for the machine vision system. A camera, located over the underlit window, is used to locate parts on the conveyor. The robot can then take parts from the conveyor for subsequent assembly. Operating the horizontal conveyor at a higher speed than the inclined conveyor helps to further singulate the parts. Parts which are in unsuitable orientations or are overlapping are dropped onto the third conveyor which returns them to the bulk hopper for re-feeding.

In the past year, the basic design has been refined and tested. A crucial aspect of the flexible parts feeder is the use of underlit translucent belts, which enable high contrast for overhead vision.

It was found that reliable vision performance required highly uniform lighting and that proper belt material selection and lighting technique are important in achieving this. We have had fair success with the use of multiple, compact fluorescent lights. This approach, however, suffers from significant background lighting non-uniformity as well as accelerated bulb failure (unless adequate cooling methods are employed to prevent overheating). Testing has shown that woven fiber-optic panels provide cooler, more reliable and more uniform back-lighting.

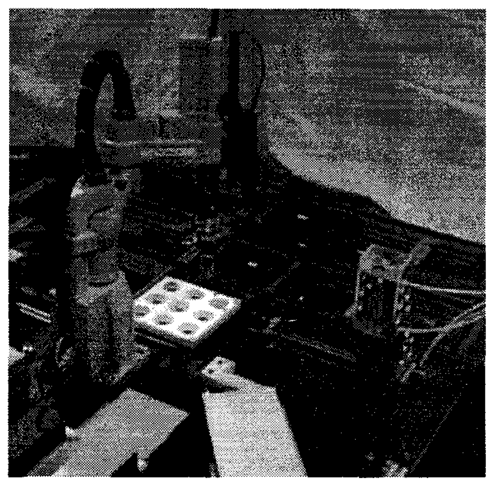

Figure 2: Assembly Station

To improve feeder efficiency, closed-loop servo control was added to the inclined and horizontal conveyors, replacing the previous open-loop speed controllers[1]. Servo control has enabled much finer regulation of the position, speed, and acceleration of the conveyor belts. This change has allowed lower-speed and reverse operation of the conveyors, controlled high speed advances, and rapid forward/reverse motions which can help further singulate parts. Such control improves the flexibility and performance of the feeders by allowing custom tuning for feeding rates, separation of parts drawn from the hopper, and more efficient coordination with the vision system, all achievable under software control and requiring no physical adjustment of the feeder.

2.2 Spurs and Worktables

In Quinn et al.[1] we introduced the concepts of "spurs" and modular worktables for flexibility and efficient use of space[4] (Figure 1). The use of spurs better exploits a robot's limited workspace, while modular worktables allow rapid change-out of dedicated tooling and fixturing. Both for pallets shunted to spurs and modular worktables swapped into pre-defined locations, precise mechanical registration is not necessary. Fiducial marks on pallets and worktables enable computation of actual coordinates using machine vision. In practice, we have verified that robot part-handling programs can be successfully adjusted automatically according to the coordinates of tables and pallets, computed from machine vision.

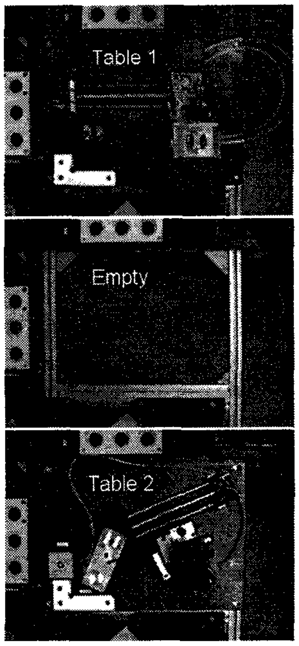

Figure 3: Worktable Changeout Sequence

A recent improvement in the modular worktables is the introduction of quick-connect pneumatic and electrical harnesses. A single pneumatic connection supplies air to a bank of solenoids mounted on the underside of the table. These solenoids control all the table hardware. A 37-pin electric plug supplies all the power and signal lines to the solenoids and sensors on each table. By conforming to a uniform wiring convention, any robot in the system can control any worktable interfaced to it using identical table-control code. Currently there are four tables in use for two different assemblies.

In practice, we have been able to change out worktables containing custom tooling and fixturing for dissimilar assembly applications, register the installed worktables, and initiate successful production of a new assembly in about 15 minutes (Figure 3). In contrast, comparable changeovers in conventional systems can take days or weeks.

2.3 Agile Grippers

Attention to gripper design is an important aspect of achieving efficient and reliable robotic part handling. To optimize throughput, gripper designs should help minimize arm motions. In our experience, automatic tool changers are useful for rapid changeover to new assemblies, but for efficient operation, tool changes should be avoided during a given assembly. Multiple grippers on a single, pneumatic rotary wrist should be utilized wherever possible to minimize arm movements and avoid tool changes. Better still, gripper finger designs that permit handling multiple parts should be sought. In our implementation, we can handle three or more distinctly different parts by a single robot without tool changes.

Use of modular wrist tool-change connectors enables our system to perform rapid changeover between assemblies. Such connectors support routing multiple independent air lines to a multi-function end effector. As with our modular tables, establishing a pneumatic circuit convention allows us to utilize any gripper cluster on any robot with identical software.

To minimize cost and complexity, we have elected to use SCARA-type robots exclusively. However, the decision to limit mobility to four degrees of freedom has important consequences for our flexible parts feeding scheme. In our feeding approach, parts are dumped randomly onto a presentation conveyor and settle to statically stable orientations. Generally, it is desirable to design the parts so that, probabilistically, a high percentage of them land in orientations that are advantageous for grasping and assembly. Unfortunately, it is often difficult to satisfy this objective. For example, a long, slender part may need to be inserted vertically into an assembly from above (e.g. a pin into a vertical hole), but it is impractical to design the pin to be stable standing on end.

Carlisle et al. developed a lightweight gripper that permits parts to rotate passively under the force of gravity without re-grasping[10]. However, some parts were found to be too light to overcome friction and had to be brushed over a fixed lip to rotate them. We solved this problem by designing a novel, lightweight pneumatic rotary-jaw gripper which actively rotates parts without re-grasping. Such capability provides the benefits of an extra wrist axis at lower cost and complexity than a servoed degree of freedom. A single pneumatic cylinder is used to drive the finger pads through a four bar mechanism. The opposing jaws of the gripper are kinematically constrained to prevent relative rotational motion, which helps assure dependable grasp during rotation. This design has a total mass of less than 7 ounces, which allows it to be used on a rotary wrist carrying multiple grippers. Figure 4 shows a part being lifted from the feeder and being rotated into an orientation needed for assembly.

A further extension of our gripper design methodology is currently under development. This new approach achieves rapid retooling of gripper fingers and fixtures through the use of rapid prototyping technology. With current technology, custom gripper-finger designs can be generated automatically from CAD descriptions of parts to be grasped. The process enables simultaneous design of parts and tooling. Further, this approach provides a formal system for developing grippers which are capable of handling multiple parts, thereby reducing the need for time-consuming tool change operations. Further details on our automated rapid tooling design and fabrication approach appear in Velasco et al. [9,11,12]

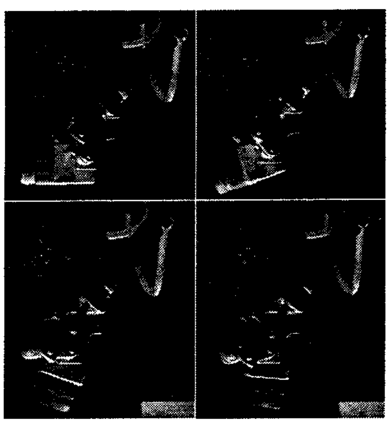

Figure 4: Rotary Jaw Gripper in Action

3. Design for Agile Manufacturability

Within the last year, we have introduced design for manufacturability considerations in simultaneous product/process design. Keeping the capabilities of an agile workcell in mind while designing new products can improve system reliability and decrease cycle-time. This concurrent engineering approach follows the work of Boothroyd and Dewhurst's Design for Manufacturing and Assembly (DFMA)[13], which espouses guidelines such as aggressively minimizing the number of components in an assembly.

Design for manufacturability teaches that the interaction of the components in a product is critical to a successful automated assembly[14]. For example, minimizing the forces required to assemble a product simplifies the needed hardware. Similarly, designing mating parts with generous tolerances and chamfers, whenever possible, permits them to be self-aligning and less sensitive to positioning inaccuracies. A few, often simple, changes to a product in the early stages of design can have a marked impact on final production[15].

In the context of agile manufacturing, some additional considerations are warranted. Parts which are to be fed in the flexible parts feeders should be designed to maximize feeder throughput. This includes consideration of all of the stable poses of a part, ideally optimizing the probability that a part will settle in an orientation well-suited for grasp and assembly by a SCARA robot. Further, one should anticipate and avoid part designs which can tangle or can damage each other during feeding, including drawing from the hopper, spilling onto the presentation surface, and returning to the hopper. In addition, parts should be considered in the context of their ease of image analysis. Since the flexible parts feeder depends on simple 2-D binary image analysis for determining part pose, it is important that parts are easily recognized in desirable poses. Care should be taken to prevent the possibility of identical silhouettes resulting from distinctly different stable orientations. This may require introducing small features or asymmetries into a part, solely for the purpose of enhancing unambiguous recognition by the vision system.

We have recently experimented with the use of rapid prototyping technology to assist the design of parts for agile manufacturing. Sample parts, designed with a CAD system, were fabricated by commercial solid freeform fabrication systems, and were tested for their feeding properties and imaging idiosyncrasies. Such experimentation with prototype parts can lead to slight design modifications that significantly improve feedability and recognizability. We anticipate that rapid prototyping will significantly assist the product/process design phase for agile manufacturing.

4. Computing Considerations for Agile Manufacturing

Integration and overall control of the agile manufacturing workcell is embodied in computer software, which is more complex than typical machine control software. The workcell is designed so that, if possible, product changes are accomplished mainly by modifying software; hardware changes are minimized to reduce cost and delay. The workcell software must be adaptable to new products without becoming unreliable or difficult to maintain. Software engineering methodology provides useful means to address these issues. The techniques of object-oriented design and programming are especially pertinent because they call for encapsulating potentially changeable design features within software modules called (object) classes. This encapsulation, or hiding, allows software maintenance to be localized. Software written using a class' services will not be invalidated by changes in their implementation, as long as the services behave as specified. However, creating a software architecture for agile manufacturing is not a straightforward application of object-oriented design. Due to the lack of published designs, substantial innovation is required.

To design software appropriate for agile manufacturing, we needed an enabling environment and definition of an appropriate software architecture. This environment required additional computing hardware, the use of commercial software tools, and simulation for software development, as described below. Definition of our software architecture was derived with consideration of both generality and efficiency.

5. Controller Design and Software Environment

Our Adept 550 robots are controlled from an Adept MV controller which consists of proprietary processors and I/O boards residing on a VMEbus. Programming requires use of Adept's V+ language and operating system. For most industrial applications, this programming environment is sufficient, but it is inadequate for an advanced agile manufacturing system. As a language, V+ is relatively primitive; as an operating system it is similarly restrictive, offering little or no support for operations such as file manipulation and task scheduling. At the initiation of this project, no proven commercial alternatives to the MV controller for controlling Adept robots[16] were known.

To overcome these obstacles, an innovative open-architecture system has been developed. Under this scheme, the MV controller is used solely for robot motion control and vision processing. All remaining functions, such as conveyor control and supervisory-level programs, are performed using third-party I/O boards and a single-board computer residing in a separate, non-Adept VMEbus.

The non-Adept computer system communicates with the MV controller through a reflective memory network[17] (essentially a block of memory shared by the two VMEbuses), sending primitive robot motion commands to the MV controller and receiving robot status from the MV controller. This controller configuration permits the use of any commercial operating system, programming language and software development tools in the non-Adept VMEbus.

The agile workcell requires more advanced multitasking features than those provided by Adept's V+. Based on agile control requirements and the advantages of using an object-oriented programming language, a commercial real-time operating system (RTOS), WindRiver's VxWorks, was chosen for the workcell controller. VxWorks provides sophisticated task control capabilities as well as task synchronization objects (e.g. System-V IPC and mutex semaphores). These features are crucial to real-time, multitasking applications such as control of manufacturing workcells where many devices operate concurrently.

VxWorks supports C++ both in the compiler suite and at the shell command-line, providing all of the benefits of object-oriented software to the embedded and real-time markets. C++ allows the software for a new assembly to reuse a large percentage of preexisting software, only writing software for the sections that are new. C++ also allows the designer to abstract the hardware details (such as how to control a particular robot) from the rest of the software, so that hardware changes (such as a robot from a different manufacturer), only affect a small section of the entire software architecture. The prototype control software for the agile workcell was written in Adept's V+. The current version is implemented in C++, executing under VxWorks.

6. Simulation in Support of Software Development

To reduce the workcell downtime required to integrate and test new software components, we have developed a 3D graphical simulator for testing and debugging the workcell control software. The software is used to control a virtual workcell (Figure 1), whose graphical representation may be observed to detect erroneous software behavior before the software is used in the actual workcell (Figure 2). Besides reducing downtime, this capability reduces the risk of injury to workcell operators and damage to hardware. Graphical simulation is especially useful for testing scenarios that are difficult to create in the real workcell, such as device collision or broken mechanisms.

In Quinn et al.[1], we reported on a conveyor simulation. More recently, we have implemented total workcell simulation[6] including robots, actuators, sensors and vision system in TELEGRIP. The workcell control program runs on a CPU board on the VxWorks VMEbus and the 3D graphical simulation runs on a Silicon Graphics Indigo[2]. A communication gateway program runs on the same platform as the simulator to drive the animation from the received commands and to feed the simulated sensor signals back to the control program. From the perspective of the control program, it is unable to distinguish between communicating with the simulator and with the real workcell. Thus the same control program can run both the simulator and the real workcell.

8. Software Architecture for Agile Manufacturing

Our approach to specifying a software architecture for the CWRU agile workcell was to identify the principal objects and classes of the workcell environment and the design patterns that interrelate them. Gamma et al define the latter as "descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context."[18] Our most important design pattern is the *Assembler-Supplier-Transporter* pattern[5], which relates the agents which assemble parts, the agents which supply parts to them, and the agent which transports parts and assemblies around the workcell. This and other design patterns abstract fundamental aspects of workcell operation and can be reused in a variety of agile manufacturing applications.

Our major classes defined in the software architecture are the assembler, the supplier and the transporter.

An *Assembler* is an agent that carries out a particular assembly sequence, with the aid of other agents. An *Assembler* directly or indirectly employs a robot, the computer vision system, parts feeders, and possibly special assembly hardware. It produces assemblies and places them on conveyor pallets for transport. It also requests parts and subassemblies from *Part Suppliers*.

A *Supplier* is an agent that is responsible for providing parts to an *Assembler*. There are essentially two kinds of *Suppliers*. A *Parts Supplier* locates a part for an *Assembler* to pick up by invoking a flexible parts feeder using computer vision. A *Parcel Supplier* obtains a fixture of subassemblies from an *Assembler* and invokes the transportation system to move them to another *Assembler* or to an unloading station.

The *Transporter* agent abstracts the underlying transportation hardware (currently a Bosch conveyor) to provide general-purpose transportation services to the workcell. It is invoked by *Parcel Suppliers* to move fixtures of assemblies from station to station.

9. Conclusions

Our research into agile manufacturing has been motivated by industrial applications and has been grounded in physical implementation. Through this empirically-driven approach, we have realized a variety of specific techniques and general principles for the design of agile manufacturing systems.

From the mechanical design perspective, the critical issue of achieving agile parts feeding can be addressed with flexible parts feeders exploiting machine vision. In addition, modular worktables and conveyor spurs are important concepts for optimizing use of valuable robot workspace, as well as supporting rapid changeover to new assembly tasks. Modular worktables and conveyor spurs also benefit from the use of machine vision to achieve the required accuracy of coordinate transforms. Lightweight, low-cost, multi-use gripper designs can offer significantly improved system throughput. Rapid prototyping technology can facilitate rapid design of multi-function grippers.

Design for manufacturing is important for reliable operation and reducing cycle time. For agile manufacturing, there are some additional considerations. Parts should be analyzed with respect to their mechanical behavior within a flexible parts feeder, as well as with respect to their ease of recognition by a machine vision system. Rapid prototyping can be exploited to test these properties physically, before design specifications are committed.

Software considerations are central to successful agile manufacturing. An object-oriented perspective is valuable for achieving generality, maintainability, and promoting ease of reprogrammability for new applications.

10. Acknowledgments

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

[1] R. D. Quinn et al. Design of an Agile Manufacturing Workcell for Light Mechanical Applications. *Proceedings of the 1996 IEEE International Conference on Robotics and Automation,* 858-863, 1996.

[2] R. D. Quinn et al. Design of an Agile Manufacturing Workcell for Light Mechanical Applications. *Video Proceedings of the 1996 IEEE International Conference on Robotics and Automation*, 1996.

[3] H. K. Rampersad. A Case Study in the Design of Flexible Assembly Systems. *The International Journal of Flexible Manufacturing Systems*, 7:255-286, 1995.

[4] R. D. Quinn et al. An Agile Manufacturing Workcell Design. *IIE Transactions on Design and Manufacturing, Special Issue on Agile Manufacturing*, in press.

[5] Y. Kim et al. A flexible software architecture for agile manufacturing. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, 1997.

[6] J. Jo et al. Virtual Testing of Agile Manufacturing Software using 3D Graphical Simulation. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation,* 1997.

[7] Greg Causey et al. Design of a Flexible Parts Feeding System. *Proceedings of the 1997 IEEE International Conference on Robotics and Automation, 1997.*

[8] R. D. Quinn et al. Advances in Agile Manufacturing. *Video Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, 1997.

[9] V. B. Velasco Jr. et al. Automated Gripper and Fixture Customization via Rapid Prototyping. *CAISR Technical Report* TR96-105, Case Western Reserve University, 1996.

[10] B. Carlisle, K. Goldberg, A. Rao, J. Wiegley. A Pivoting Gripper for Feeding Industrial Parts. *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1650-1655, 1994.

[11] V. B. Velasco Jr. et al. Computer-Assisted Gripper and Fixture Customization via Rapid Prototyping. Submitted to IASTED *Fifth International Conference on Robotics and Manufacturing*.

[12] V. B. Velasco Jr. and W. S. Newman. An approach to automated gripper customization using rapid prototyping technology. *CAISR Technical Report* TR-96-101, Case Western Reserve University, 1996.

[13] G. Boothroyd. Product design for manufacture and assembly. *Computer-Aided Design*, 26:505-520, 1994.

[14] A.J. Scarr, D.H. Jackson, R.S. McMaster. Product Design for Robotic and Automated Assembly. *Proceedings of the 1986 IEEE International Conference on Robotics and Automation, 796-801, 1986.*

[15] G. Kovacs. Changing Paradigms in Manufacturing Automation. *IEEE International Conference on Robotics and Automation Proceedings*, 3343-3348, 1996.

[16] Open Architecture: Myth or Pipe Dream? *Robotics World*, 14(1):18-20, Spring 1996.

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

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