To appear in the Proceedings of the
8^{th} Internation Symposium of Robotics Research
October 4-7, 1997 Hayama Japan

Opportunities for Increased Intelligence and Autonomy in Robotic Systems for Manufacturing

Alfred A. Rizzi and Ralph L. Hollis
in collaboration with
Ben Brown, Zack Butler, Greg Fries, Jay Gowdy, Wing-Choi Ma,
Patrick Muir, Arthur Quaid and Xing-Xing Yu

The Robotics Institute
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213 USA

URL: http//www.cs.cmu.edu/~msl

1 Introduction

Since the introduction of programmable industrial robots at the beginning of the 1970s, an industry based on these machines grew steadily to a point in the mid-1980s where it was poised for an explosion of huge proportions. This explosion failed to happen as a "robotics backlash" took hold, with the strong perception held by many that these machines did not, and would not, live up to the high expectations held by their users.

While the interest in robotics by industry waned, it continued to grow tremendously within the academic community and many difficult and fundamental problem areas began to be investigated. The emphasis has been on key components, (*e.g.* kinematics, dynamics, control, 3D vision, planning, etc.) but not complete systems, and integration of these results into industrial practice has been slow.

One result of this history, in the opinion of the authors, is that academic robotics researchers—perhaps faced with the difficulties of technology transfer to industry—have "turned their backs" on industrial robotics in favor of working in other, more exploratory, areas such as mobile, field, medical, space, and service robotics. Meanwhile, worldwide sales of industrial robots has steadily grown during the past decade. To enable further growth, there must be significant changes in the way robotic systems are deployed in manufacturing environments. It would appear that there are now significant opportunities for applying increased *intelligence* and *autonomy* to industrial robot systems. This observation derives from several factors, including:

(i) Increasing demand for ever smaller and more complex products whose lifetimes are ever shortening.

(ii) The need to remove humans from the immediate vicinity of the manufacturing process because of scale and cleanliness requirements.

(iii) The recent widespread and ubiquitous availability of significant computing power at reasonable costs.

(iv) The Internet explosion.

1.1 Practical Difficulties with Automated Manufacturing Systems

The fundamental problem with "modern" robotic manufacturing systems, is that the individual components (robots, part-feeders, conveyor systems, etc.) are generally designed as stand-alone devices. As a result, little or no explicit effort is dedicated to enabling the integration of these factory components into a complete manufacturing system. Similarly, information about the design of most complete systems is scarce, as there are few incentives for system integrators to document their work, and it is difficult to extract durable truths from case studies. There remain prohibitively high economic and technical costs associated with the factory integration process that in turn severely limit the utilization of robotic elements in many practical applications.

Meanwhile, work on programming robotic assembly systems has progressed at both the task-level (*e.g.* "place part A on part B") and at the manipulator level (*e.g.* "move joint 3 10.15 inches; close gripper; etc."). To date, task-level systems have not moved out of the laboratory, while the manipulator-level systems, despite their enormous programming complexities, have become widely accepted.

1.2 Related Efforts

A number of academic and industrial groups have attempted to provide partial solutions to these fundamental problems over the past decade.

A leading-edge benchmark, which attempts to address some of these issues, is Sony's SMART flexible assembly line. It makes use of SCARA robots equipped with indexing multiple grippers and modular product and part transport systems to simplify the mechanical problems of factory reconfiguration. Unfortunately the individual modules are physically large, and the problems of programming and tuning a complete factory system are still daunting.

A key study was the DARPA microfactory demonstration [4], developed as part of the Defense Department's Intelligent Task Automation program. This work emphasized operation in unstructured environments, recognition and grasping of overlapping parts, semiautomatic planning, and geometric reasoning. The system which resulted used parts kitting, and sensor-moderated motion to assemble a precision microswitch. While meeting many objectives, the system required 18 minutes to complete the microswitch assembly task.

A recent significant trend is the notion of programming and operating robots over the Internet. For example, a concept of virtual laboratories was recently demonstrated [5], showing that a robot in one laboratory can be programmed and controlled from another laboratory thousands of miles away. In another case, exploration and tele-gardening [6] was demonstrated. Both of these studies show that it is now possible to allow meaningful remote (Internet based) interaction between a robot and a programmer or operator. However, neither demonstrates the level of expressiveness required to undertake a significant practical manufacturing task.

Sandia National Laboratory has developed its "Agile Manufacturing Prototyping System," (AMPS) comprised of robotic cells supplied by various vendors[1]. Simultaneously, industrial robot producers have begun to service the demand for increased flexibility and precision. Adept has developed a concept of "rapid deployment automation" [3] that embraces key elements of modularity and off-line programming. Megamation and Yaskawa have produced systems of small, modular, easily programmed robots capable of moderately precise assembly.

Our vision of an agile manufacturing system is one that provides for a large pallet of modular robotic processing and product transport systems from a wide variety of vendors; with each module presenting a standard mechanical, computational, and algorithmic interface enabling their simple and rapid integration (both physical, and programatic) into a complete factory system. In contrast to much "academic" research on agile manufacturing systems, we are not striving to provide a "universal assembly machine," but rather wish to adhere to the industrially accepted model of flow-through (assembly line) processing while providing for the rapid deployment and reconfiguration of such systems. We forsee this being achieved through the use of compact, mechanically simple elements whose customizable combined behaviors provides the specific complex capabilities required for a specific application. Furthermore, we do not foresee these modules being used to form a "lights-out" factory, but rather we expect them to act smoothly in concert with humans, serving as highly capable and intelligent tools for their operators.

---

[1] See http://www.sandia.gov/AMPSfact.html

2 The Agile Assembly Architecture

As part of a multi-million dollar four-year project funded under the NSF Multi-Disciplinary Challenges component of the High Performance Computing and Communication program, we are developing new hardware and software technologies and strategies for automated assembly of precision high-value products such as magnetic storage devices, palm-top and wearable computers, and other high-density equipment [7]. We envision a design cycle focused on the development of virtual factories which could combine resources for producing products from geographically distant locations. Our approach draws extensively on high-speed wide-area communication and intensive distributed computation. The Agile Assembly Architecture (AAA) supports the creation of miniature manufacturing systems (minifactories), built from small modular robotic components, which will occupy drastically less floor space than today's automated assembly lines. Our goals are to reduce assembly system changeover times, facilitate geographically distributed design and deployment of assembly system, and to increase product quality levels.

We are developing AAA as a distributed system of tightly integrated mechanical and computational robotic modules endowed not only with information about their own capabilities but also with the ability to appreciate their role in the factory as a whole and negotiate with their peers in order to participate in flexible factory-level cooperation [12]. A unified interface tool will allow a user to select and order these mechanisms over the Internet and to assemble, program, and monitor them in both a simulated factory environment and the real factory environment.

AAA relies on factory-wide standard procedures and protocols, and well-structured autonomy to simplify the process of designing and programming high-precision distributed assembly systems. The architecture makes use of modules' self knowledge and ability to explore their environments to make the transition between simulation and reality as painless and seamless as possible.

Our sample instantiation of these ideas is a modular tabletop factory that we refer to as *minifactory*. The key technical ideas include the use of distributed low-DOF robotic modules or agents[2], and integrated product transport and manipulation subsystems. The entire system is composed of compact elements with standardized mechanical and electrical interconnects allowing for the rapid setup and adjustment of a factory system.

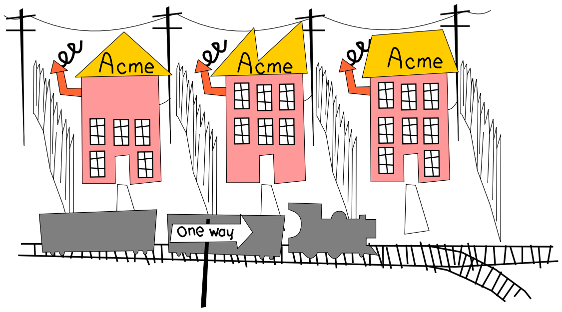

Figure 1: Cartoon characterization of a state of the art modular manufacturing system.

Figure 1 depicts a cartoon chacterization of what we would consider a state of the art modular manufacturing system. Whereas this approach is widely regarded as a significant advance beyond older non-modular systems, there are significant characteristics which limit its agility. In the figure, the large and fairly complex factory modules (represented by buildings) all tend to be produced by a single vendor, and are separated from one another by fairly rigid interfaces (the high fences) greatly limiting flexibility. Furthermore, each module requires a semi-custom interface (walkway) to the essentially inflexible product transport system (railway). Conveyors belts and similar mechanisms capable of only one-way flow are the norm in such systems. This makes for a system which can be very efficient (haul a lot of freight) in production, but which is difficult to change.

Modules tend to be of fixed size, independent of the functionality provided, leading to inefficient use of floor space. If changing market conditions require modifications to the manufacturing system the factory designer is faced with the choice of either modifying the internal functions of modules or adding/subtracting modules. In the first

---

[2] By agent we explicitly mean a mechanically, computational, and algorithmically modular manufacturing entity, *e.g.* robot, capable of both communication and physical interaction with its peers.

case, altering the module severely interferes with its modularity and can be complicated by the module's inherent complexity. In the second case, inserting or deleting modules requires either making space by physically moving all upstream or downstream modules, or closing up gaps created when modules are removed.

Finally, it is extremely difficult for more than one module vendor to participate in this scheme. Because of module complexity, the cost of entry is high. There is a "winner take all" force at work here where there is every incentive for a given vendor to offer only a "complete" line of modules (even if some may not employ the best technology available), and really little incentive for a company which has only the capability to field a few modules to participate in a modular system controlled by another company. It can be argued that this situation leads to inefficiencies, less than optimum performance, and poor agility with respect to the marketplace.

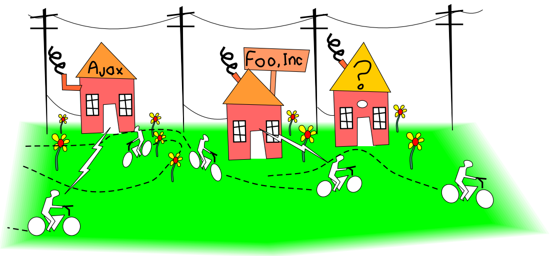

Figure 2: Cartoon characterization of an AAA based manufacturing system.

In contrast to this highly structured model of factory automation, Figure 2 depicts a similar rendition of a manufacturing system based on the ideas of the AAA. Here, the factory processing elements (large regularly placed and isolated tenements) have been replaced by a collection of modules (small cottages) from a variety of vendors, which are placed as needed in the manufacturing system. The high fences between these modules have been removed since each module is designed with the explicit intent of interacting with their neighbors. Finally the product transport system (centralized railroad) has been replaced by a large collection of moving modules (bicycle couriers). These capable and independent agents, who explicitly coordinate their actions with both the factory processing elements (home owners) and each other both form a highly flexibly product transport system and can participate in local manipulation and processing tasks. This approach is less efficient than the state of the art systems in production, but on the other hand, it is much easier to set up and change.

Adding or deleting functionality is relatively straightforward in this approach. New modules can be added to the system almost anywhere (picture additional cottages placed in front, back, or side yards with perhaps some small adjustments to the positions of the surrounding cottages), and taking modules away will have no effect on the existing ones. We propose that a module's internal functionality never be subject to modification by a factory designer, but rather only by the module's vendor. Adhering to this stricture ensures that modularity, and hence the system's inherent agility, will not be broken.

Because each factory module is fairly simple and limited in functionality, the cost of entry for module suppliers is kept low. In this way, each modular component can employ the latest and best technology from that particular company. We argue that this sort of approach can respond more quickly to changing market opportunities than can today's state of the art modular systems.

2.1 Underlying Challenges

A fundamental component of our long term goal is to elevate the design of automated assembly systems from the detailed technical problems associated with designing and integrating independent mechanisms to the more salient problem of designing the factory as a whole. We see this as a complimentary effort to that provided by the industrial engineering and operations research communities, but one that can provide a natural mechanism for the widespread application of new factory design methodologies. Ideally the collections of machines that follow the framework presented here will form a natural template onto which the more abstract ideas of factory design and optimization can be applied.

From our perspective there are several key barriers that currently stand between current best practice and a more agile and open manufacturing infrastructure. Fundamentally these barriers all relate to the need for standard mechanisms to support interaction between agents, designers attempting to integrate the agents into a system, programmers developing control software for a factory involving the agents, and operators whose task is to monitor performance and provide support when an agent is unable to cope with its environment.

As mentioned in Section 1 the current practice in the robotics and automation community is to focus on the engineering of individual robots and mechanisms, with little or no consideration for how they are later integrated into a complete factory system. Only by designing robotic modules that are explicitly prepared to participate in a larger factory system can we begin to provide the types of tools necessary to move towards fundamentally more useful systems of machines. This represents what we see to be an important goal of the robotics field: the construction of mechanisms capable of both physical and "social" interaction. Physical manipulation has been the province of both the academic and industrial robotics communities from their inception. While "social" interaction has been an academic goal [9] which has produced several novel and interesting systems [1, 8]. There have been, however, few practical applications for these systems.

We believe our efforts to design rapidly reconfigurable and "user friendly" factories represents a modest step towards achieving this goal. The scope of "social" interaction has been explicitly limited to two well defined domains: inter-agent interaction for factory coordination, and interaction between agents and the factory personnel. Given even this limited scope, the details underlying the definition of these agent interfaces are not obvious and comprises a significant portion of the Agile Assembly Architecture.

2.1.1 *Factory Interaction*

The definition of a suitable "machine language," for inter-agent communication is a central issue in enabling the type of interaction under discussion between both multiple robotic agents, and humans and agents. For inter-agent communication the basic requirements include:

Extensibility: Whatever the actual format of messages, the underlying media must efficiently allow for the introduction of not only new message formats, but also the negotiation of completely separate communication modalities. In principle, these allow the natural growth and development of new methods for inter-agent coordination, and with responsible classification of which protocols are to be considered "required" and which are "optional" for an agent it is reasonable to expect long term compatibility through use of the required basic protocols.

Real-Time Coordination: Sufficient communications capability (enough bandwidth with sufficiently low latency) is essential to allow the tight coupling and integration of agents which are incapable of performing complete manufacturing tasks in isolation. In the sample system described in Section 3 it is clear that there are significant advantages both in terms of flexibility and simplicity inherent in supporting such distributed mechanisms. This issue is mitigated by the fact that in general an individual member of a factory is likely to only interact with a well defined "neighborhood" of peers—*e.g.* an insertion mechanism need only perform precise coordination with a part feeder (providing the part to insert) and with a product transport mechanism (presenting the subassembly to be operated on)—greatly limiting the scope of high performance communication by nature of its locality.

Factory Communication: Conversely, there is a need to provide a standard means for factory wide control and monitoring, and hence the need for a standard interface to join every robotic module in a factory together with each other and an arbitrary number of control and monitoring workstations. The intent is to provide a common medium and basic interchange format for the most rudimentary forms of factory control while simultaneously providing a means by which modules can negotiate for the use of more application specific interchange formats. By requiring every element of a factory system to "understand" this basic level of interaction, we strive to ensure that each and every machine is capable of participating in the factory at a minimum level.

2.1.2 *Integrated Design, Simulation, and Evaluation Tools*

Not only is it necessary to require a facility for interaction between machines, it is equally important to support interaction between humans acting as factory designers and the agents. In a traditional design process there are three major classes of interaction between the designer and a component under consideration.

(i) **Preliminary selection:** Initial evaluation of a manufacturing component for its suitability to a problem.

(ii) **Detailed evaluation:** Iterative validation and discard of candidate solutions and components based on analysis, simulation, and mock-up of proposed designs.

(iii) **Integration and refinement:** Detailed analysis, design, construction, and test of a working system.

While the distinctions are somewhat arbitrary, they highlight fundamentally different forms of inquiry performed on candidate components by a designer, and provide a model under which we can explore the interactions necessary to reduce the designers' uncertainty about the factory system they are working on.

Given the widespread acceptance and rapid development of high-performance computation and communication systems, particularly as embodied in the Internet, we foresee the integration of such capabilities with robotic agents as enabling a new kind of relationship between the designer and the component. In stage (i) of the design process where traditionally decisions would be made based primarily on vendor-provided catalogs, it now becomes possible for the designers (or some agent acting on their behalf) to directly interrogate an actual mechanism (probably at a vendors facility) for relevant properties. There are a myriad of options for exactly what remote entity answers such queries depending on the nature of the component under scrutiny. In the case of "brainless" components this would be similar to a catalog search, but for full-fledged factory agents the designer could interact directly with the specific machine under consideration, potentially providing a significantly more accurate representation of the actual mechanism and its capabilities.

The implications of this model on phase (ii) of the design process are more significant. It now becomes possible for the component under evaluation to provide a number of different "renderings" of its physical and behavioral models for use by a designer. With the careful integration of tools for either retrieving down-loadable representations from factory components, or remotely involving the component in a distributed simulation or analysis process it becomes possible for the item under review to provide a model with an appropriate level of fidelity to its actual performance. It is easy to imagine a broad range of models ranging from trivial kinematic representations to highly detailed physics-based distributed simulations or even remote experimental environments being made available to a designer through a single and consistent set of design and simulation tools capable of allowing the construction and interrogation of a highly accurate "virtual" factory identical to the design under evaluation.

Finally, in phase (iii) as a design is refined and physical experiments are undertaken, it is through these same tools that the designer will continue to interact with the evolving factory design. The ultimate goal being the truly seamless transition from factory simulation to operation, but with enough expressiveness and flexibility in the underlying components and representations so as not to unduly constrain the behavior and performance of the final system.

2.1.3 *Programming Interaction*

The key goal in simplifying human/machine interaction is one of providing a simple and natural language for specifying complete machine behavior. Further complicating the problem is that given a large collection of disparate agents, it will be necessary to distribute the "factory program" among the various agents. However, in contrast to current practice, it will be necessary to provide tools and highly expressive, yet convenient, languages that aid a factory programmer in developing and debugging the agent level programs which instantiate a specific solution to a manufacturing problem. Most of our effort in this domain is being directed at understanding and developing appropriate representations for machine behavior in a factory setting—hoping to take advantage of the rapidly developing field of human computer interaction to provide flexible and expressive interfaces between our representations and the factory programmers. We feel this aspect of the larger problem is most suitable for immediate investigation, as it is the natural avenue through which to explore the advantages of increased autonomy on the part of the individual robotic agents, while still being closely related to the more well understood problems of assembly planning and factory optimization.

2.1.4 *Operator Interaction*

Finally, as we do not see the near-term future of automated manufacturing to be “lights-out,” it is important to consider the role played by factory operators and their interaction with the agents that make up the factory. Predominantly we see operators serving as aids to the factory, acting to help agents recover from and avoid situations that they are unable to manage in an automatic manner. This includes such mundane tasks as managing factory supplies by refilling part supplies and removing finished products, possibly for additional processing by a more traditional factory system. Furthermore, we foresee operators being called to the aid of agents that recognize factory difficulties that they are unable to recover from. This form of interaction should include the ability of an agent to notify an operator of the difficulty, allowing the operator to remotely (from across the room or facility) interact with the agent in question and its peers via a set of agent “front-panels” or “dashboards” — remotely rendered presentations of the agent’s status — to diagnose the problem and choose a corrective course of action.

3 Instantiation

We are currently designing and building a working example of a modular tabletop factory or *minifactory* conforming to our notion of the agile assembly architecture. This minifactory incorporates planar robot couriers that travel on connected tabletop platen surfaces. These robots are derived from planar linear motors that float on air bearings and translate along the platens in two directions with micron-level precision. Other devices, including 2-DOF “overhead manipulators,” are mounted on modular bridges above the platens (see Figure 3). The couriers are responsible for both carrying the product subassemblies from one overhead device to another, and cooperating with the overhead devices to execute assembly operations. Limiting the robots to 2 DOF has advantages in terms of modularity, reliability, and performance [11], but allows the minifactory to perform 4-DOF assembly tasks through the use of robot cooperation. Each low-DOF device will have integrated high-performance computing resources, and serve as an agent in the AAA context. This eliminates the need for central resources that would degrade the modularity and scalability of the system.

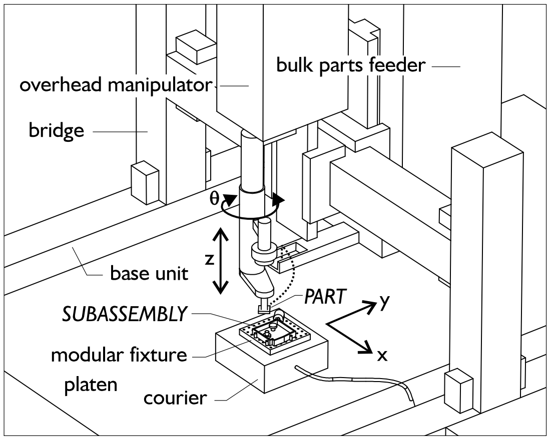

Figure 3: Basic components of a minifactory.

We are currently developing the key electromechanical elements, including couriers, manipulators, precision parts feeders, and other modular components. We are implementing a distributed realtime computer architecture, modeling and simulation software, high-level network communication protocol, and graphical programming tools to support the long-term vision. We believe this type of system can only be developed through careful integration of hardware and software tools in a manner heretofore unseen in the robotics and automation community.

3.1 What is a Minifactory?

In addition to the limitations on the forms of interaction between factory elements implied by Section 2, we have deliberately chosen to restrict the scope of mechanical capabilities we wish a minifactory to perform to afford both analytic tractability and design practicality. Toward this end we have limited the class of tasks to assembly and processing operations requiring four or fewer degrees of freedom. Specifically we want to construct systems capable of:

- Four-degree-of-freedom vertical insertion.
- Easily integrating overhead processing (*e.g.* laser processing or material/glue deposition).

- Micron-level part placement accuracy.
- Factory design and programming in less than a week.

To provide this functionality, a minifactory consists of a potentially large collection of mechanically, computationally, and algorithmically distributed modules. Each element in this collection is responsible for providing a minimum level of cooperation and communication in order to participate in the most basic minifactory operations.

The most obvious departure from traditional automation systems and one of the most obvious embodiments of our philosophy of factory level integration can be seen in our choice to integrate product transfer and local manipulation. As such, we have eschewed the traditional use of SCARA manipulators coupled with part conveyor systems and local fixtures. Alternatively, as depicted in Figure 3, we have chosen to make use of two-DOF manipulators and two-DOF planar couriers moving over a high-precision platen surface. The couriers are thus responsible both for product transport within the factory and for transiently forming cooperative four-DOF manipulators when they present sub-assemblies to a stationary manipulator.

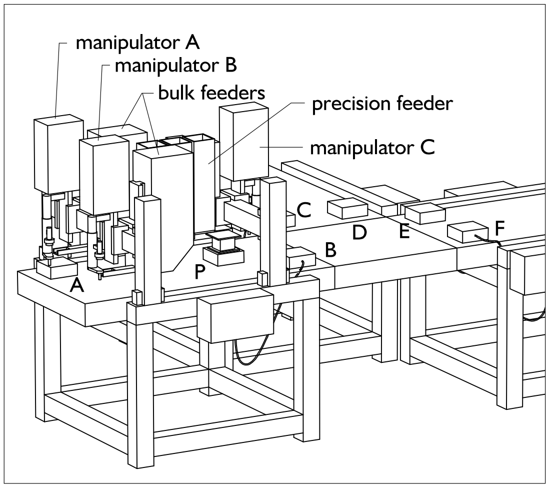

Figure 4: View of a "typical" section of a minifactory assembly system, including a "tee" junction.

Perhaps the best way to appreciate the implications of this approach to the design of factory level assembly systems is to consider a somewhat contrived but illustrative example. Figure 4 depicts a view of a small section of a fictitious minifactory. The system pictured includes six couriers and three manipulators with two bulk-random parts feeders and one precision feeder. Couriers begin on the left of this system, present their sub-assemblies to the first two manipulators where two new components are added; the resultant sub-assemblies then travel to the right where the final manipulator is responsible for both placing a precision component and transferring the final assembly to one of four couriers.

3.2 Run-Time Coordination and Communication

Any element of a minifactory, be it a courier, a manipulator, or some custom designed module, must provide a minimal level of capability in order to participate in the minifactory "society." Currently we foresee there being three general classes of capability every agent must reliably provide: *basic trustworthiness*, *self initialization*, and *inter-agent coordination*, the latter which includes facilities for resource negotiation.

3.2.1 *Basic Trustworthiness*

For an agent to be a successful member of a factory community its peers must be able to trust it to reliably represent itself. Practically, we see this manifesting itself in the form of three fundamental capabilities.

- All agents must advertise their basic capabilities and the protocols they understand to their peers.
- Every agent must be capable of reporting its current status and its understanding of its environment.
- Each agent must implement reliable and safe failure detection and recovery schemes.

The first two of these requirements are essential to address the issues of Section 2.1.1 and support the graceful coordination between minifactory components, their peers, and factory monitoring tools. Furthermore, the ability to advertise capabilities addresses the need for a predefined extensible protocol suitable for the exchange of such information between agents. The next two capabilities may well be the most important, and quite possibly the most difficult to precisely define and implement. The assertions demand that agents be capable of constantly monitoring the state of the factory available to them. Furthermore, when an agent detects conditions outside the norm it must be capable of independently correcting the aberration, negotiating with its peers to recover from the fault, or broadcasting its inability to proceed thus bringing the factory to an orderly stop. Although it is potentially difficult to guarantee this level of capability in an arbitrary system we feel that through judicious use of a combination of traditional AI reasoning [10] and reactive behaviors [2] that it can be achieved in the highly-constrained domain of minifactory.

3.2.2 *Factory Calibration/Initialization*

Integral to the rapid deployment of an assembly system is the need for precise and automatic calibration and initialization whenever a factory is "turned on." There are three interrelated tasks that must be collectively undertaken by the minifactory components to successfully initialize a factory system. This process will begin with agents identifying their peers through the use of messages broadcast to the factory at large. Following this, couriers must explore their environs to discover both the exact geometry of the platen surfaces, as well as the positions of any stationary agents within their range of motion. Finally, through a careful exchange of this information between agents, a complete map of the minifactory can be constructed both in the agents and in a monitoring interface tool.

3.2.3 *Robotic Agent Coordination*

Since individual elements of the minifactory are rarely capable of performing "useful" tasks alone, it is essential to include standard mechanisms for orchestrating their coordination. Fortunately, the locality of action performed by individual agents provides a natural locality of communication and coordination. For example, manipulators A and B in Figure 4 both only interact with couriers A and B, while couriers C through F interact with everything other than manipulators A and B. To help alleviate the problems associated with manually coordinating the motions of all of these machines, we have chosen to make use of a *geometry reservation system*. Under this system, factory elements which are potential competitors for a specific predefined segment of space are grouped and required to negotiate for the use of that shared resource. In principle, an individual machine may well be a member of several different "groups of agents" sharing a myriad of resources associated not only with physical resources but potentially with more abstract factory goals. It is the neighborhood groupings of machines that form the basic fabric for cooperation between the elements of the factory system.

The most fundamental form of this cooperation will happen whenever a courier and manipulator transiently form a four-degree-of-freedom system to perform a part placement task. The most basic mode for such cooperation will take the form of a virtual linkage between two machines where one agent is effectively slaved to the state of the other, allowing for simple coordinated movement. Other modes of cooperation will include coordinated behavior changes and cooperative sensor-based action. Behavior changes will be used to encode the sequence of operations necessary for a high-precision force-controlled insertion task (*e.g.* manipulator exerts low vertical force while the courier "finds" the hole, followed by the courier becoming compliant while the manipulator exerts higher forces to perform the insertion).

3.3 User-Level Design, Programming, and Monitoring Tools

In the absence of a centralized controller, a minifactory will have a centralized *user interface tool* capable of supporting the design, simulation, and run-time monitoring and control of a minifactory. Each element of a minifactory—whether it be a courier, manipulator, or other custom robot—is an independent computational entity. The overall behavior of the minifactory results from the interaction between these elements and their environments. The central challenge for the minifactory simulation and programming environment is to provide the services of Section 2.1.2, facilitating the development of well-debugged distributed programs, while simultaneously easing the difficult transition from the simulated world of bytes and pixels to the real world of actuators and sensors.

3.3.1 *Design and Programming*

The goal of the minifactory programming environment is to simplify the difficult problem of selecting and integrating the components of a factory while generating the distributed programs for every agent in the system. The desired outcome is that a minifactory system will be able to be designed and programmed by an expert in the domain of the assembly problem at hand without requiring expertise in "minifactory programming."

Within this framework we foresee the use of constraints, such as local frames of reference, and abstractions, such as coordinating the gross motion of couriers through the use of distributed resource management rather than considering it as a global allocation problem. Such uses of constraints guides the user to construct robust systems while abstractions hide details that the user cannot afford to be concerned with if correct programs are to be rapidly built.

This approach to geometry management and gross motion planning provides several advantages to the user:

- **Abstraction:** The user does not have to specify that a specific courier must explicitly contact some other courier and/or manipulator for permission to move into a manipulator's workspace; all of these negotiations are hidden through the use of a reservation area.
- **Modularity:** Rather than a manipulator knowing it has to interact with a particular courier, it just has to know that it interacts with whatever courier has reserved its workspace.
- **Robustness:** Since the reservation areas are referenced to physical components of the factory, if these components move slightly, the various agent programs will continue to function properly.

3.3.2 *Simulation*

Minifactory simulations allow a user to explore the application of minifactory technology to a particular assembly problem, and to do much of the development and debugging of the factory programs off line in a virtual environment. A key to a minifactory's rapid and successful deployment is the nearly seamless transition of a factory program from this virtual environment to the actual machines. The two facets of our architecture that make this transition possible are *fidelity* of the simulation and *robustness* of the underlying agents. Fidelity demands that the simulated factory will behave reasonably closely to a physical system under similar conditions. Robustness of the agents acknowledges the inability to configure an actual factory in exactly the same manner as it was simulated, but that the differences can be detected and accounted for.

The issue of fidelity is addressed through an agent's self knowledge and self representation. Each minifactory module provides a representation of its own geometry, behavior, and integration constraints. Thus, the simulator will not use a catalog to look up the characteristics of a typical manipulator, but rather will query an actual manipulator via an Internet connection for its own self representation. This reliable representation of an agent's characteristics eliminates many inaccuracies that would otherwise occur. Robustness is provided through the inclusion of additional sensing resources that enable the individual agents to self-calibrate and explore their environment (as described in Section 3.2.2).

Additionally we foresee simplifying the transition from simulation to reality by allowing mixed operation of the simulation system in conjunction with running hardware. In full simulation mode, most of the agent models and programs will be internal to the simulation environment itself, each of them having been constructed from the description provided by the agents themselves. In practice there is no reason beyond efficiency why the implementation of these agent models could not be performed by the remote agents themselves rather than internal to the simulator. Thus, simply by mixing internal agent models and external agents, a real agent could be put through its paces in isolation, with all its actuators working and sensors gathering data, but within the context of a greater simulated system.

4 Conclusion

The AAA project began in November, 1995. So far, we have concentrated most of our efforts on developing the engineering technologies needed to build the prototype minifactory. We are also well into the tasks of developing comprehensive environments for modeling, simulation, and programming. A prototype 3D interactive user interface package has been implemented which allows the user to look at a detailed running minifactory simulation with zoom, pan, and other controls. This factory simulation can be downloaded from our web site and viewed by anyone. Many of the AAA design, operation, and agent coordination issues have been formulated [12].

Looking to the future, we foresee AAA and minifactory serving both as a research testbed and as an exemplar of what we see as one potential path for the future of automated assembly systems. Whereas our focus has been on rapidly reconfigurable assembly systems for precision assembly, this approach may also have utility for agile parts fabrication, chemical synthesis, pharmaceutical manufacturing, and other such applications.

We are encouraged by the positive response received to date from our colleagues in both the research and the industrial communities. In particular, it would seem the notions of increasing both *intelligence* and *autonomy* through the use of well-crafted modular building building blocks to realize rapid deployment is very attractive—provided that it can be made to work in the real world, and that it can provide realizable benefits in the marketplace.

It is our hope that solving many of the complex problems faced by modern manufacturers will be attractive enough to receive increased attention from robotics researchers, thereby re-establishing more active collaboration between these communities.

Acknowledgments

This work is supported in part by NSF grants DMI-9527190, DMI-9523156, and CDA-9503992. The authors would like to thank Arthur Quaid for providing figures 3 and 4.

References

[1] R.A. Brooks and L.A. Stein. Building brains for bodies. *Autonomous Robots*, 1(1), 1994.

[2] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. submitted to the International Journal of Robotics Reaserch, 1996.

[3] John J. Craig. Simulation-based robot cell design in AdeptRapid. In *Proceedings IEEE International Conference on Robotics and Automation*, pages 3214–3219, 1997.

[4] W. Foslien and V. Nibbe. A robotic workcell for small-batch assembly. *Robotics Today*, pages 1–5, 2nd quarter 1990.

[5] M. W. Gertz, D. B. Stewart, B. J. Nelson, and P. K. Khosla. Using hypermedia and reconfigurable software assembly to support virtual laboratories and factories. In *Proc. 5th Int'l Symp. on Robotics and Manufacturing*, Maui, Hawaii, August 15-17 1994.

[6] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Desktop teleoperation via the world wide web. In *Proceedings IEEE International Conference on Robotics and Automation*, 1995.

[7] R. L. Hollis and A. Quaid. An architecture for agile assembly. In *Proc. Am. Soc. of Precision Engineering, 10th Annual Mtg.*, Austin, TX, October 15-19 1995.

[8] D.B. Lenat. Cyc: a large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), Nov. 1995.

[9] Marvin Minsky. *The Society of Mind.* Simon and Schuster, 1986.

[10] D. J. Musliner, E. H. Durfee, and K. G. Shin. World modeling for the dynamic construction of real-time control plans. *Artificial Intelligence*, 74(1):83–127, March 1995.

[11] A. Quaid and R. L. Hollis. Cooperative 2-DOF robots for precision assembly. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, Minneapolis, May 1996.

[12] A. A. Rizzi, J. Gowdy, and R. L. Hollis. Agile assembly architecture: An agent-based approach to modular precision assembly systems. In *IEEE Int'l. Conf. on Robotics and Automation*, Albuquerque, April 1997.