

A FLEXIBLE STRUCTURE FOR COMPUTER-CONTROLLED MANUFACTURING SYSTEMS

David M. Upton

Go to start of paper

Table of Contents

Abstract

1. Introduction

1.1 Traditional FMS

1.2 FMS as a panacea for small-batch machining

1.3 The significance of FMS in the 1990s

1.3.1 Narrow Process Focus

1.3.2 Technological uncertainty

1.3.3 All-or-Nothing

1.3.4 Productivity(4)

1.3.5 Shallow learning curve

1.3.6 Level of Investment

2. FMS as a strategic cul de sac

2.1 Inflexible Flexible Manufacturing Systems

2.1.1 Failure and recovery as sources of complexity

2.2 The demise of monolithic FMS?

3. An Alternative Approach

3.1 The Computerized Product in a Society of Machines

4. The Advantages of Heterarchical control

4.1 Short-term flexibility

4.2 Long-term Flexibility

4.3 Recovery from Failure

4.4 Machine Failure

4.5 Human Intervention

4.6 The Learning Product

4.6.1 The learning product in a dynamic system

4.6.2 Difference between this and conventional FMS "learning".

4.7 Changing the Rules in the Control Room

- [4.8 Learning by Management](#)
- [4.9 Data Collection for Management Control](#)
 - [4.9.1 Managing Bottlenecks](#)
 - [4.9.2 Managing Priority Jobs](#)
 - [4.9.2.1 Grand Auction Reversal](#)
 - [4.9.2.2 Local Auction Reversal](#)
 - [4.9.2.3 Renegade machines](#)
- [**5. Enabling Technologies**](#)
 - [5.1 Intelligent Parts](#)
 - [5.2 Radios](#)
 - [5.3 Cheap computers](#)
 - [5.4 Advances in process planning](#)
- [**6. Conclusions**](#)
- [**7. Experimental Results**](#)
 - [7.1 Utilization](#)
- [**8. Performance and Routing Flexibility**](#)
 - [8.1 Tolerance of Failures](#)
 - [8.2 Resource requirements](#)
 - [8.3 Further considerations](#)
 - [8.3.1 Product limitations](#)
 - [8.3.2 Process decomposition](#)
 - [8.3.3 Sequentiality](#)
- [**9. Acknowledgments**](#)
- [**APPENDIX: The Generation of Random Manufacturing Systems**](#)

A FLEXIBLE STRUCTURE FOR COMPUTER-CONTROLLED MANUFACTURING SYSTEMS

David M Upton

Graduate School of Business Administration, Harvard University, Boston, MA 02163

Internet: upton@stig.harvard.edu

Copyright (c) 1994 Harvard Business School

A version of this paper appeared in *Manufacturing Review* Volume 5 (1), 1992 pp. 58-74.

Abstract

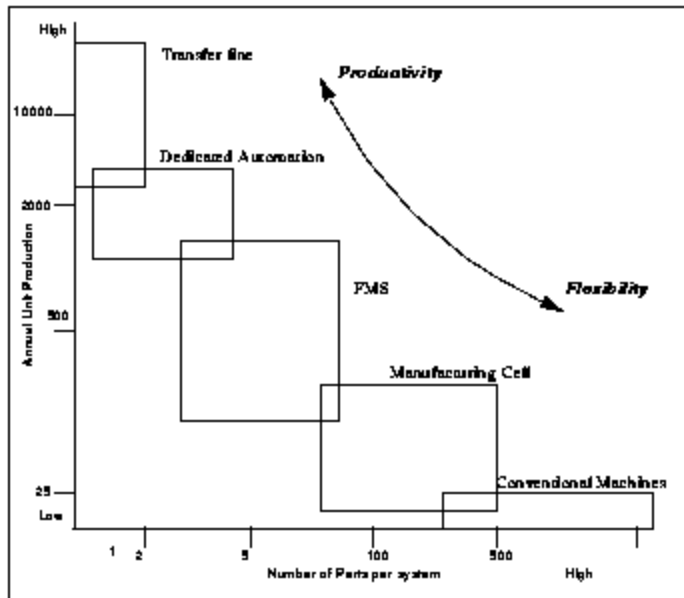
This paper describes a structure for distributed manufacturing system control, in which the product may progressively learn about the system which makes it. The strategic motivation for the work is presented along with a selection of research results exploring the general applicability of distributed methods for manufacturing system orchestration.

1. Introduction

Flexibility is one of the benefits of small-batch manufacturing. While a small-batch shop may produce lower unit output than a shop dedicated to one or two lines, its strength is that it can make a variety of different products in small volumes. The *complexity* of coordinating manual small-batch production had, until the early 1980s, confined automation of the manufacturing system as a whole to industries producing in large-batches, with a small, slowly-changing range of products. Small-batch production relied on stand-alone processing machines, which were coordinated by human operators and schedulers. The complex nature of producing a wide-range of products brought what were seen as necessary evils accommodated in the name of flexible manufacturing. If a company was machining earthmover axle components or valve housings in small batches then high inventory, unpredictable, long lead times and quality problems were very common. Manufacturing engineers such as Theo Williamson [\[1\]](#) in the 1960's were inspired by the idea of being able to bring the controllable advantages of the transfer line to the more complicated world of small-batch machining manufacture. This was an important problem: 75% of the value of items produced in US engineering firms was (and is) made in batches of 50 or less [\[2\]](#).

By the 1970's, the advent of Computer Numerically Controlled (CNC) machine tools had made the process of machining both automatic and flexible. CNC machine tools could be programmed, locally, with a method of making a component. One merely had to load a casting onto a fixture, supply an appropriate program and tooling, and the product would be predictably produced time after time. Williamson's contribution was to suggest that the coordination of the flow of jobs *between* machines could also be carried out automatically. He envisioned and built a rudimentary system (Molin's System 24) which comprised a group of CNC machine tools connected by an automatic material handling system. A centralized computer-control system oversaw the shop and coordinated and scheduled the flow of jobs between the machines. With the further advances in computer technology, and the stabilization of CNC technology, the early 1980s saw a flurry of installations of systems designed along the lines of Williamson's System 24. Pioneering companies such as Caterpillar and John Deere in the US started to build large systems which went against traditional manufacturing dogma - systems which combined economies of scope *and* scale. These large computer-controlled systems had a relatively high aggregate output yet were flexible, since they could produce a number of different products (See [Figure 1](#)).

Figure 1 The Best of Both Worlds([1](#))



Flexible Manufacturing Systems (FMS), as they were called, became a great focus of attention in industry and in academic research for a number of years [3],[4],[5],[6]. Although the more skeptical might say that behind the rapid growth of publicity and interest in FMS lay a bubble inflated by a sales-hungry machine-tool industry, it was nevertheless clear that the systems demonstrated a significant technical advance in manufacturing practice. The real strength of these FMS lay in the fact that they brought tremendous benefits in inventory reduction (often 85%), quality improvement and lead time. In many installations, the inventory reduction alone was sufficient to justify the investment in hardware, software and system design effort.

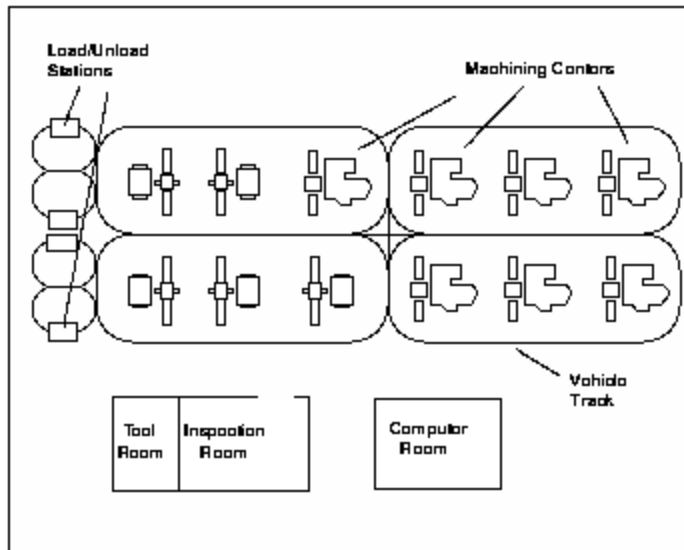
1.1 Traditional FMS

A flexible manufacturing system (FMS) is an arrangement of machines.... interconnected by a transport system. The transporter carries work to the machines on pallets or other interface units so that work-machine registration is accurate, rapid and automatic. A central computer controls both machines and transport system... National Bureau of Standards(2)

The key idea in FMS is that the co-ordination of the flow of work is carried out by a central control computer. This computer performs functions such as:

- Scheduling jobs onto the machine tools
- Downloading part-programs (giving detailed instructions on how to produce a part) to the machines.
- Sending instructions to the automated vehicle system for transportation

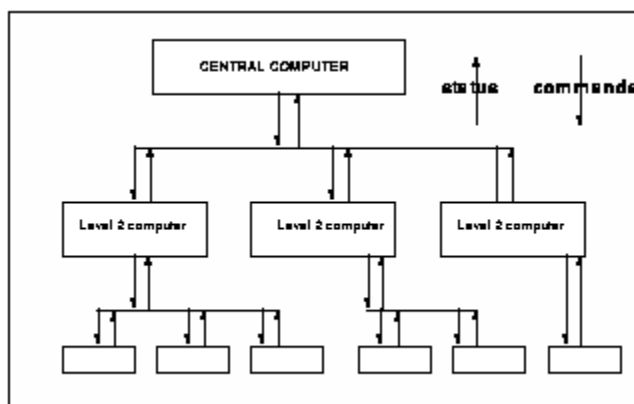
Figure 2 A Flexible Manufacturing System



Products to be produced are manually loaded onto pallets at a load station, and the computer system takes over, moving the product to the various processing stations using automatic vehicles, which may be rail-guided, guided by wires embedded in the floor or free-roving. After having visited all necessary stations, usually only two or three, the job is taken back to the load station, where it is removed from the pallet and passed to the next process.

Writing FMS control software is not a trivial matter. The software is often custom-written, and is not straightforward programming task. There are complex, real-time interactions with remote hardware which require great expertise and experience on the part of the programmer, particularly for larger systems. In order to simplify this problem, many systems use a hierarchical approach to real-time control [7], [8],[9]. Each computer controls a team of underlings, collecting status reports and issuing commands. Commands flow down the hierarchy, while status reports flow up.

Figure 3 Hierarchical Control



Various estimates are that between 20% and 40% of the cost of FMS installations are in computer software and hardware development.

1.2 FMS as a panacea for small-batch machining

While the interest in FMS as a solution to the problem of automating the job-shop was growing rapidly in the early 1980s, some researchers began to urge caution. Jaikumar's work [10] showed a marked difference between the US and Japan in the advantage taken of the flexibility possible with FMS. Far from exploiting the possibilities of the technology, US manufacturers were using FMS as fixed lines which happened to produce a small group of products, rather than to provide versatility and mutability after they were installed. The management of flexibility was poorly understood. Another study carried out by a UK consulting firm reported that a small group of companies estimated that they had achieved 40% of the benefits of FMS before any hardware was installed or a line of software written [11],[12]. The reasons for this became clear as more companies experienced the phenomenon. In order to automate shop coordination, the company needed to understand it, and formalize a solution to the problems in the shop. This process was itself a very valuable one. The coordination system had to be brought "under control". Having understood the needs, rationalized the products being made in the shop, and understanding the coordination system well enough to avoid obvious inefficiencies, the advantages of computerizing the shop became less significant. Even so, there is no evidence to suggest that this 40% was achievable, let alone sustainable, without a Flexible Manufacturing System actually being installed in the plant. The advantages of a well-run FMS were clear: short lead-times, low inventory and a step towards the factory of the future.

1.3 The significance of FMS in the 1990s

The installed worldwide FMS base in 1989 was estimated to be around be between 500 [13] and 1200 systems [14], the higher figure arising when a system is defined as having 2 or more CNC machine tools connected by a materials handling system, and controlled by a central computer. Ranta and Tchijov [14] suggest that this number will rise to around 2500-3500 by the year 2000. This led them to suggest that *"the strategic majority of production of the metal-working industries in the industrialized countries will be produced by FMS or similar systems [by the year 2000]."*

Kelley's empirical research in 1987 strongly contradicts this. In a large (>1000 firms) survey of US metal-working firms, she found that less than 5% of those plants *with computerized automation* have an FMS and that FMS constituted only 1.5% of the total number of installations of computerized automation [15]. Why are there still so few FMS in the world given that small-batch engineering production is a significant proportion of manufacturing output? There are significant practical reasons for the disparity between the promise of FMS in the 1980s and its narrowness and scarcity of application in the early 1990s. These reasons are outlined below separately, though they are very much interdependent.

1.3.1 Narrow Process Focus

The types of manufacturing processes suitable for integration into traditional FMS remain limited: turning, milling and sheet metal work dominate FMS processes while many other, less well automated, processes remain unintegrated. This is mainly because they are not computerized at the machine level and are hence not yet ready for computer integration at the system level. Nevertheless, even in metal *cutting*, with much wider application of Computer Numerical Control, comparatively little output is due to FMS.

1.3.2 Technological uncertainty

When FMS were first introduced, the novelty of the integration technology naturally made many firms "wait-and-see" until the technology had settled. This was particularly true in the smaller companies. The technology of FMS has, at least in the West, not become mature and well understood and many companies would still consider FMS technology to be "high-risk".

1.3.3 All-or-Nothing

The monolithic all-or-nothing nature of FMS increases the risk of projects, causing companies to shy away. This is particularly true of those companies whose products are a little different from those for which FMS has already proven itself(3) --- the scale of the effort required, in conjunction with their less standard processes is sufficient to dissuade them from undertaking the project.

1.3.4 Productivity(4)

In many applications, the productivity of the prospective system --- in terms of its output with respect to its capital input --- is insufficient. Practical experience has also shown that the utilization of the systems may be much lower than predicted when they were designed, further reducing productivity. While productivity may not be the manufacturing performance criterion most closely associated with the competitive focus of the system, there are bare minima to be exceeded in any industry. Without a reasonable level of practical productivity (and hence return) from capital, the project will founder, perhaps rightly, in the capital investment procedures of the firm.

1.3.5 Shallow learning curve

It takes a long time for an organization to learn about FMS technology. Much of the technology is embodied in software integration, and software engineering is not a skill which many manufacturing companies acquire quickly.

Second, the highly interdependent and specialized nature of the technology means that integration is best handled by a very tight nucleus of people . While this might get the job done at the outset (once these scarce people have been found), it often means that just a few people hold the key competencies. This concentration of knowledge inhibits learning in the organization as a whole.

The nature of the skills required means that these skilled people have often been imported from outside the firm and owe it only fleeting allegiance. When they leave, they take their skills with them, which further flattens the learning curve of the company.

1.3.6 Level of Investment

The investment in FMS (as characterized by Ingersoll Engineers [12]) is often in the range of \$10 to 15 million. The amount of money needed to finance an FMS is thus a significant barrier to its introduction, particularly in smaller companies. Smaller firms currently perform most of the small batch work, so it is here where FMS would be most appropriate. However, for most small firms, an investment in FMS would mean "betting the farm". Quite reasonably, given the plethora of other difficulties, they choose not to.

These six reasons, in concert, marshal against the diffusion of current FMS technology. This is not to say however, that these are sound reasons why FMS should not be embraced. Many argue that the difficulties described above are the price one has to pay, and that technologies such as FMS must be seen as a *strategic* investment --- the short-term hurdles must be compared against the long-term strategic and intangible cost of being ignorant of the technology. If this argument were truly compelling, one might expect many more of the forward-thinking companies, whose competitiveness is tightly linked to their small-batch effectiveness to have grasped the nettle, and to have adopted FMS technology as a stepping-stone towards the future factory and as a strategic investment in the flexible technology of the 21st-century plant [17].

This is not the case. There are foreboding reasons (inherent in the existing technology) why current FMS, even when justified on strategic grounds, simply do not make sense.

2. FMS as a strategic cul de sac

Despite many claims that FMS investment should be viewed as a strategic investment in flexibility, I will argue that FMS as they are currently structured, are often characterized by a common feature: their *inflexibility*.

2.1 Inflexible Flexible Manufacturing Systems

The main disadvantage with FMS technology lies, paradoxically, in its inflexibility. FMS are flexible in that they can, in the short-term, produce a range of known products⁽⁵⁾. However, the complexity necessary to automatically achieve short-term flexibility makes it difficult to introduce new families of products into the system, and certainly much more difficult than in a manual shop. Similarly, when new machines are to be added (or old ones updated) it can be very costly. Changes in system configuration require time-consuming, expensive alteration to software ^{[19], [20]}, particularly in complex, Western systems.

IIASA's⁽⁶⁾ FMS database shows that successful systems with payback in less than 5 years are either

- very small (< 4 CNC machines) and simple

or are

- very large (> 15 machines) and complex.

Simple systems work because they may be committed to a focussed group of components, and be relatively unsophisticated in their use of software. Large systems work because the cost of the control software may be distributed over more output. Even the large systems, however, show unimpressive returns given the risks involved, and only moderate ability to introduce new products. Only 18 of the 800 FMS in IIASA's sample belong to this category.

The software for controlling medium/large FMS has to handle the tremendous complexity of scheduling and dispatching multiple products through a variety of processing routes, transporting them around the system and recovering from any failures in system components. This cleverness means very complex software. The complexity of the software necessary to provide short-term flexibility has frequently become the millstone which constrains long-term flexibility.

While companies may have an idea of the *functions* of products they might be producing in five years' time, they may be unable to guarantee that the components in those products conform to a specific engineering family, and this is what FMS often currently demands. If new products need to be introduced, and the manufacturing system equipped with new machines here and there, FMS will exact a high fee, particularly for firms without sophisticated expertise. Research has found that the considerable expertise assembled to install the original system has often dwindled by the time it is necessary to re-configure it ^[21].

For new families of products and the addition of new machines, FMS simply costs a lot to change: surely a definition of inflexibility rather than flexibility. This means that companies are less nimble in chasing changing markets with new product ranges.

2.1.1 Failure and recovery as sources of complexity

In a complex, automatic system like an FMS, many failures must be anticipated and catered for. For example, tool breakage, and machine and vehicle failure are fairly common. In order to deal with

such circumstances, methods for recovering from the failure must be built into the central control software. This has a number of effects.

- The control software becomes more complex, further limiting long-term flexibility.
- Any changes to the software also demand similar recovery procedures built into them, making the changes even more expensive.
- In order to run automatically, the system often has to use many sensors so that the control computer can deduce the failure state of the system and can recover from it.

Despite all the effort put into the software, the sheer complexity of running a small-batch shop means that the system programs cannot anticipate all possible failure states, which means that manual intervention is inevitable. It is clear that a centralized, all-seeing, all-knowing control computer is not possible --and the closer systems come to having one, the more complex (and less long-term flexible) they become. The software becomes hardware.

2.2 The demise of monolithic FMS?

The evidence paints a bleak, inflexible picture of FMS. In spite of the occasional well-publicized exception, the promise of the 1980s has not been fulfilled. Many companies have seen their inflexible white-elephants incur tremendous costs, as markets shifted and their "flexible" manufacturing system was found to be unable to accommodate the change [22]. This is reflected in the market. Sales of FMS, particularly large, complex systems with many machine tools, have slowed markedly, even when one takes into account the recession. The reasons for this are slowly becoming clear: as a strategic investment in automated flexibility, FMS as they are currently conceived are anything but.

3. An Alternative Approach

One solution to this problem (and one which many companies have adopted) is to accept the need for long-term flexibility in their manufacturing system, and provide it with manual, skilled labor and CNC machines. This is a very flexible system. Just-in-time methods, Group Technology (GT) and manufacturing cells all help the manual small-batch shop combat the tendency to disorganized chaos which is often seen in the industry. These techniques are all effective methods of managing the flow of work around the shop, especially for smaller parts. However, this solution essentially abandons automation, and demands that people continue to attend to the processes required to make the part, though they may attend to multiple machines. How might one free people to do more meaningful work, than attending to machines which are, nowadays, largely automatic? How might one go about controlling or *orchestrating* a large (say 50) congregation of CNC machine tools without the long-term inflexibility of the traditional FMS, and without the need for human intervention?

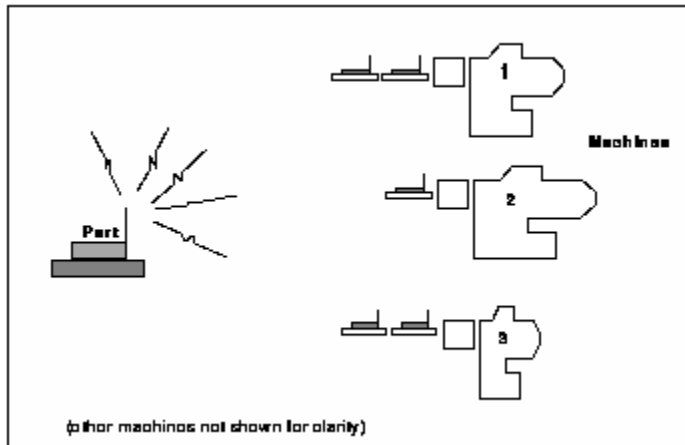
The following section describes one alternative approach to a hierarchical FMS for a machining system.

3.1 The Computerized Product in a Society of Machines

Consider a machine shop with fifty CNC machine tools and automatic vehicles for transportation of materials. This section outlines a new type of control structure for such a system. It is best outlined by describing an example of the production procedure which is followed in order to produce an individual item. A production-control computer requests that a casting which has arrived at the shop be machined. The casting is manually bolted in a flexible fixture on a pallet, and a small computer is fitted to the pallet. This computer contains a processor, some memory and a radio. This assembly will be referred to as "the part". The production control system loads the memory of the part's computer with the processing requirements of the product. In other words, the control system tells the part

what it needs to look like. The product enters a short system entry buffer, and while it is waiting, its computer broadcasts its description throughout the system to the flexible CNC machine tools. Some machines examine the job's description and decide that they are unable to machine the product because it is too big for their bed, say. Others simply have the wrong type of geometry for the job. Still others decide that they *can* do the processing work on the product. The machines' computers plan a process for the job, and decide how long it would take them (and how much it might cost in terms of tool-wear) to do the work.

Figure 4 Part transmits processing request and data to the system.



Having determined how much processing time is needed, the machine checks its local buffer, determines how many jobs it has waiting for it, and forms a "bid" on the job. It transmits this bid across the network to the waiting part. The part waits until a system-set deadline to receive bids and having collected them, selects a *winner* from the bidding machines. It sends a message to that machine that it has selected it and expects to arrive for processing. The next arrangement the part needs to make is for transportation to the machine. Here there are a number of possibilities but let us say, for now, that the part essentially "calls a cab": It requests transportation from an automated vehicle dispatcher which sends a vehicle to it, and it arrives at the machine. After waiting in line it is machined. While it is waiting it arranges subsequent machining and ultimately leaves the system, once all tasks are complete. Having been processed the part moves out of the system to be assembled into its final product.

The part has thus been processed *without* a central control computer, and with simple, modular, physically decentralized hardware and software.

4. The Advantages of Heterarchical control

The benefits of such a *heterarchical* approach to system control are quite wide-ranging. The control method obviates the need for an on-line central control computer and the accompanying software. A number of authors describe variations of this basic approach to a variety of control problems [23], [24], [25], [26], [27], [28], [29], [30], [31]. Pioneering manufacturing work was carried out by Duffie [32] and Parunak [33]. Shaw explores the scheduling advantages of distributed control over its centralized counterpart [34]. An excellent review is provided by Dilts et al. [35]. In the system described in this paper, each machine is equipped with an on-board process planner, which determines how *that machine* will process the job. This allows a wide variety of machines to contribute to the capacity of the system. The machines also have a local (cheap) bidding

microcomputer which is interfaced to the process planner and with its communicator. These may be retrofitted to CNC machines which are already in the company.

A shop run like this reflects the nature of manual job-shops much more closely than a rigid hierarchy of computers. While the original attempts at hierarchical control attempted to mirror the hierarchical function of a manual shop (foreman → chargehand → operator), they ignored the fact that the most decisions in a job-shop are made by interaction and negotiation within a hierarchical level. The shop described here more closely resembles a manual job-shop run in a Just-In-Time fashion.

4.1 Short-term flexibility

In order to make FMS control feasible, many companies split their manufacturing systems into manageable "chunks" of 7-12 CNC machines and integrate these into an FMS. In manual production, machine shops of 1000 machines are not unusual. An often quoted figure is that 1 CNC machine-tool can match the capacity of 5-10 manually controlled, stand alone machine tools. Where then are the FMS equivalents of the large shops, with 100 or 200 CNC machines integrated together? The answer is that such a shop would not be economically controllable with traditional, centralized FMS. Few companies would have the skills available to put one together as a monolith. In any case, the poor long-term flexibility caused by the complexity needed for the system would make it financially and strategically unattractive. For this reason, FMS are limited in size by their controllability and poor flexibility. However, in a distributed system like this, there need be no such size constraints and the part may avail itself of a much larger group of machines, not just the one which it has access to in its traditionally controllable chunk. The fact that the part has access to more machines, and that no group of components is specified in advance for those machines to make, means that many different parts can be serviced appropriately by the system.

The relative advantage here depends to a large extent on the degree of multiple redundancy in the system as a whole. Such redundancy is becoming more prevalent as a result of technological changes in machine tools. Today's CNC machines are considerably more versatile than they were 15 years ago. The move from 3-axis to 5 and 6 axis machines, the use of modular tooling systems and pallet-changers along with the improvement of tool-management systems have dramatically increased the scope of jobs which an individual machine can undertake. This trend is likely to continue as machine tool manufacturers accommodate more and more operations into one workpiece setting and one machine tool. An allied trend is the increase in information processing ability at the machine level. The continuing technological advances in physical versatility and in the distribution of computing in machine tools imply increased multiple redundancy in systems, as well as local, rather than central, information processing.

4.2 Long-term Flexibility

Consider the software changes necessary in order to add a machine to this system. An important question is how long the manufacturing system has to be down in order to implement the software changes. In this system, there is no need to take the system down. A new machine may simply be told the "rules of the game", and be installed in the plant with power and access to tooling. It becomes part of the system with no lost production. Removing a machine from the system is also straightforward. The machine simply stops bidding, and jobs stop coming to it.

Each addition and removal has very limited system-wide ramifications. The machines rise and fall in utilization as they become more or less appropriate for the current product range. This greatly facilitates new product and process introductions, since any peculiar requirements for processing of a new product may be introduced without systemic disruption.

4.3 Recovery from Failure

The system is particularly graceful in recovering from failures. This is because it does not rely on a centralized computer which may need to scramble to find out the failure state, or be told the failure state by a human operator. Failures are limited to the locale where they occur and system-wide consequences are avoided.

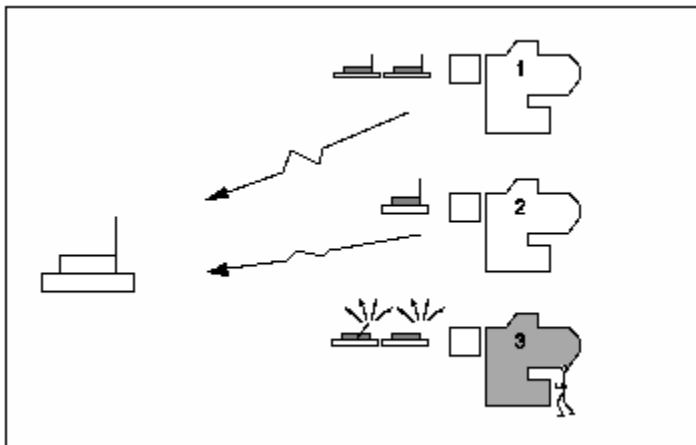
4.4 Machine Failure

If a machine fails, a number of local actions may be taken which avoid the need for complex centralized control software and hardware. Machines are given the rule:

If you are defunct, do not bid

This is fortunately the default for machines whose computers are too defunct to use the rule and avoids the problem of parts being assigned to machines which have failed (see [Figure 5](#)).

Figure 5 Machines transmit an estimate to the job. In this case, machine 3 is down, and fails to respond to the request.



Maintenance staff can repair the machines, and then simply allow them to start bidding again. Temporary removal and re-introduction of the machine does not require system-wide shut down procedures. Parts which are "stranded" (that is, are in the input buffer of a failed machine), have a rule which says:

If you are waiting for a machine for more than twice what you expected to wait, re-instigate the bidding procedure and find another machine.

The part will find another taker and be transported from the input buffer without human intervention. It can reinstate the auction process and arrange to have itself removed from the carousel. The system may thus recover without more far-reaching consequences or the need for central software to be written to deal with the situation. The part, with its simple, generic, software can arrange its own recovery.

4.5 Human Intervention

In a monolithic control system, humans are easily excluded from the manufacturing system for a number of reasons: If a human being changes the state of the system, then that person needs to either

tell the central control computer or that computer needs to have sensors to detect the change. Such a change might include taking down an individual machine for preventative maintenance. This often means that only people with an understanding of the system as a whole can be involved, or at least, that systems engineering "experts" are around to ensure that the system-wide consequences of local actions are anticipated.

A "society" of machines such as this, with distributed orchestration ensures that the system is accessible to people who are not systems engineers. Provided people understand the basic, local rules which machines and parts obey, then they will quickly be able to foresee the consequences of any local action. Since recovery is built into the system through its distributed nature, they need not worry that these local actions will bring the whole system down.

4.6 The Learning Product

Once a part has been machined, its computer will retain a record of its processing experience. This record will maintain such information as

Machine identifying itself as "Number 4" made the following features <feature list>. Machine bid 25 minutes, actually took 45 minutes. Suspect machine failure.
or

Machine 17 failed to complete promised operations. Found successful alternative in machine 42. The part will thus keep a record of its history. The part may relinquish its memory /computer so that other products coming through the system may make use of it, and draw inferences from the experiences of previous parts processed by the system. For example,

Machine 7 appears to underestimate its processing time. Make correction of +25% to any bids.
or

Machine 14 has a probability of failure of .07 for any job it undertakes. While the details would depend on the actual application, the idea is simple. The parts should have a collective memory(7), which grows and allows the system to learn about itself, and improve its own performance. The fact that the parts' memory is attached to a succession of multiple bodies (parts) does not affect this idea of progressive learning (see [Figure 6](#)).

4.6.1 The learning product in a dynamic system

An important issue here is the provision of a mechanism for forgetting. There need to be a method by which the information which the parts learn may progressively forgotten to take into account system changes, and the repair of chronic unreliability problems. There are two mechanisms by which this make take place.

First, we may build "forgetfulness" into the system functionality. For example, if a part has just finished processing, and needs to update the current predictor of some system parameter which it has experienced, say the fractional error in processing time estimates of machine 24, $\hat{\tau}_{24}$. If this part's experience of this parameter is τ , then the new estimate of the parameter, $\hat{\tau}_{24}^+$ may be generated by setting

$$\hat{\tau}_{24}^+ = (\alpha - 1) \hat{\tau}_{24} + \alpha \tau$$

where $\alpha (0 < \alpha < 1)$ is a parameter reflecting the "forgetfulness" of the system.(8).

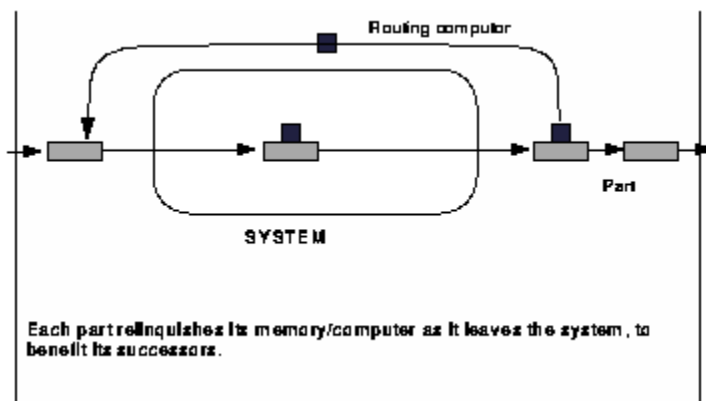
Second, we may intervene as managers of the collective memory. When a persistently faulty machine is corrected, we may wish to step in and deliberately erase all memory of the miscreant device . Parts may then begin to learn afresh from new samples, rather than slowly forgetting the now unrepresentative experience --- we are thus forcing the parts to forgive quicker than they otherwise might. Ways in which such global changes are effected on the distributed system are discussed in Section 4.7 below.

4.6.2 Difference between this and conventional FMS "learning".

In a conventional FMS, the complexity of the control software act as an inhibitor which resists change in the system, and hence learning. In traditional FMS, which are architecturally static, managers must rely on humans to notice systemic problems and to risk opening a Pandora's box of software to remedy them. The distributed method described above learns experientially and acts on that knowledge autonomously. Learning is an inherent feature of the architecture.

There is much work to be done in exploring this idea --- the conditions for effective and efficient learning, along with the development of criteria to determine which factors are best learned and which are best "told" are important avenues of research. Work is currently being carried out in these areas --- in particular, in the application of automata theory to the problem (viewing the parts as learning stochastic automata).

Figure 6 The Collective Memory



4.7 Changing the Rules in the Control Room

While the idea of allowing entities to negotiate in order to orchestrate the system is attractive, we still need some method of changing the behaviour of the system in order to effect "control" at a higher level. Control in the system described above is effected by using *procedural* rather than *substantive* rules.

The system as it is described attracts a variety of metaphors. A useful one is that of the system which gets commuters to and from work by automobile in a large city every day. There is no central controller or hierarchy of computers governing the operation of the system at a micro-level with commands such as:

Car 43 turn left at 6.17 am

Instead, each entity in the system is endowed with a modicum of intelligence and has a goal with some rules.

get to work but obey the rules of the road

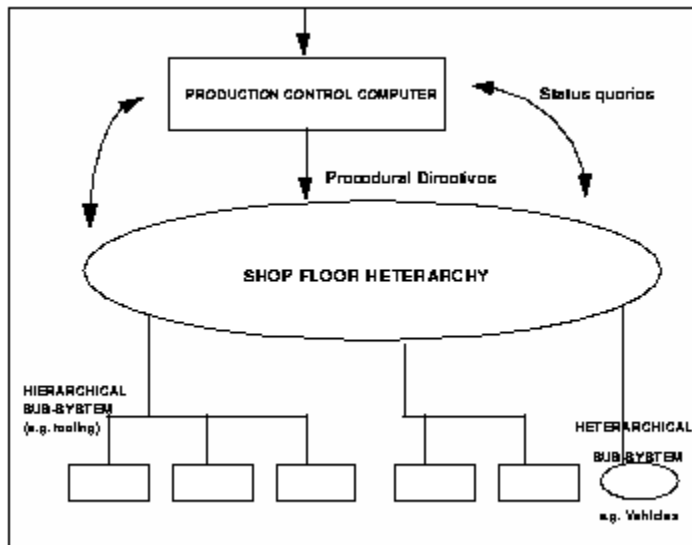
After that, control is effected by using *procedural* rules. These might include one-way streets or traffic lights.

4.8 Learning by Management

In the manufacturing system, we may change the rules which the various entities use when negotiating with each other. For example, we may decide to allow a machine to pick priority parts out of its buffer if their priority number is high enough (see . This, of course, will mean that commitments made to previous parts will have been broken. Nevertheless, this is one control choice which can be made in a straightforward way. Changes in procedural rules may be put into operation by transmitting new rules to the entity computers (machines, parts) across the network. These rule changes should be infrequent, and should be made after very careful consideration and simulation by the system managers. It is straightforward for the company to progressively learn the best methods for controlling the shop by experimentation and to embody these methods in the distributed rule-base.

While the computers on the floor do not operate hierarchically, the heterarchy is nevertheless embedded in a broader corporate/plant hierarchy. The shop simply behaves as a different kind of subordinate: as a group of processors using robust, distributed rules to perform rather than a group obeying direct commands passed down a bureaucracy from a dictatorial central computer.

Figure 7 Hybrid Control (heterarchy within a hierarchy)



4.9 Data Collection for Management Control

In a traditional FMS, data about parts is collected by the central control computer whether it is needed or not. Hopefully, when it is needed it is up-to-date and accurate - unfortunately, manufacturing is not a deterministic activity.

This system does not, in general, collect data as a matter of course. If information about a job is required, the system is interrogated in real-time across the network. An example of such a question is:

Part 17843, Order 2873. Where are you? How long do you expect to take to be completed?
The production control computer might need to know this in order to update an MRP system running in the plant (as an example only). The part may reply:

At least 38 minutes. I am currently at Machine 4, which bid 17 minutes on the current processes. I then need to be processed by Machine 17 which took 21 minutes the last time it processed a part of my type. It does have a history of taking a mean of 3 minutes longer than it estimates.
(Natural language is used for communication where possible to increase the transparency of the system for its operators). The principle is that data is not collected and stored only to go out of date or not be required. A query system is provided, and the information collected in real-time, from the entities closest to the action (as shown in [Figure 7](#)).

4.9.1 Managing Bottlenecks

Most manufacturing systems are characterized by having some processes which can be handled by only one or two machines [37]. Despite the redundant nature of the machines in this system, there will still be processes which may only be effected by a small group of machines, or even just one machine. In a highly dynamic system, with multiple redundancy, such capacity bottlenecks will not be constant in either severity or location, since the processing requirements and mix of jobs changes over time. In addition, such bottlenecks may not be on the ultimate processing path for all jobs. While any manufacturing system must have a bottleneck, the managerial problem concerns identifying those bottlenecks which cause a high degree of imbalance of workload between machines and ensuring that jobs with a unique requirement for the bottlenecks are served in preference to other jobs which have alternatives available. For this reason, we may divide bottlenecks into two classes for the system described here: global and specific.

Global bottlenecks are those processors which currently limit the capacity of the system as a whole. These bottlenecks may be identified by asking machines (by broadcast message) to report their current utilization. It is then up to the system managers to decide whether they wish to relieve the bottleneck by supplying another suitably equipped machine.

Specific bottlenecks apply more severely to a particular group of jobs, and grow more troublesome as the demand for those jobs increases. In order that these jobs are not further delayed at their specific bottlenecks (due to jobs which could have gone to some other machine), machines may have a rule which says (for example):

If your utilization is more than 90%, confine your bidding to jobs which have been rejected by the system more than 15 times in the last 5 minutes.

In this way, machines may intelligently pick up orphan parts, and ignore jobs which can find alternatives processors, regardless of how effective they are in producing those parts.

4.9.2 Managing Priority Jobs

It is also necessary to provide a mechanism for priority jobs, and allow them to jump ahead of jobs which have already arranged processing. Methods for providing this capability as an integral part of the structure described above are the subject of an ongoing research project. Some of the specific techniques are outlined in general terms below:

4.9.2.1 Grand Auction Reversal

When the system becomes very busy, and there are a large number of priority "grades" among the parts, we may allow the direction of auction to change throughout system. That is, machines no longer bid on jobs, rather, jobs bid on machines. Thus, the customers become the servers, and jobs bid for machine time with some function of their priority as currency. In these circumstances, all entities must recognize conditions which "flip" the system into this new state. The key issues here are:

- the development of criteria for reversing auction direction: for example, number of late jobs, global average machine utilization, utilization of bottlenecks.
- the distributed management of the transition state
- the exploration of auction protocols for this "reversed" state - this reversed state may be indeed be the "normal" state for some systems.

4.9.2.2 Local Auction Reversal

Similarly, there are methods by which an individual job may pre-empt machines involved in the "normal" bidding. Jobs may be tagged with a *trump card*, giving machines permission to bid on them, while ignoring existing pending jobs, and allowing them to bring the job to the front of their processing queue if they are successful in winning the auction. This is one of the simpler solutions to the priority problem, particularly in cases when there are only two classes of job (urgent and regular).

4.9.2.3 Renegade machines

We may allow only a small group of the machines in the community to renege on contracts they have previously arranged. The provision of a group of renegade machines permits the swift processing of urgent jobs, but allows the system as a whole to avoid continually evaluating the prospect of breaking existing contracts.

As the above research continues, there will emerge many opportunities for operations researchers. A very small part of the vast existing operations research literature on scheduling and dispatch is useful here. The reason that most of the work is not useful, is that operations research models of scheduling and dispatch usually assume the existence of a centralized being (computer/human) whose commands the system obeys. We have removed this object, for the sake of producing a long-term competitive manufacturing system. This means however, that there are a number of practical challenges for the OR community working in manufacturing: the practicable integration of effective heuristics with the control architecture which must effect them.

5. Enabling Technologies

There are some infrastructural elements which are essential in order to allow a system like this to work. Much of this infra-structure is a result of on-going research in a variety of technologies.

Various associated technological advances are also needed, in addition to those required for infrastructure.

5.1 Intelligent Parts

The idea of putting memory into parts is not new, and it is certainly not very difficult to also endow them with some processing capability. A small computer, with the ability to carry part-descriptions in static RAM is the minimum requirement. Otto Bilz in Germany already have a system for cutting

tools which allows the tool to carry its offset information electronically and communicate by UHF radio to the machine tool on which it is being used.

5.2 Radios

Cheap radio communications systems for entities in the system are essential. These must have access to a network which allows both broadcast and point-to-point messages, and allows seamless addition/removal of nodes without central control. A number of network protocols now have this capability.

5.3 Cheap computers

In order for each job in the system to have its own computer, computers need to be relatively cheap. For some products, it may be worth leaving its manufacturing computer embedded in it for the rest of its life so that manufacturing information may be retrieved at some time in the future for quality control or replacement reasons. A copy of the parts memory will, of course, have been made in order to perform the learning functions described on [page 13](#).

5.4 Advances in process planning

The automatic generation of process plans for parts is currently feasible only in manufacturing research laboratories for general products. Headway has been made in research and industry in order to make this a possibility.

The important thing is that a part's description of itself and its processing requirements should not be so specific as to pre-select the machine which will ultimately make part of it, but also not be so general (a full 3D description for example) that exceptional local machine process planners are required and the data carried by the part is too expansive. The dramatic increase in the flexibility of CNC machine tools is, however, making more "generic" part programs increasingly practicable.

6. Conclusions

A more detailed discussion of the research results is presented in Section [7](#). Research is currently underway exploring such issues as learning, auction-reversal (where, in a very busy system, parts begin to bid on machines rather than vice-versa) [\[38\]](#), the control of tooling [\[39\]](#) and the effect of various communications protocols. Research on the system described here has been carried out in both discrete-event and object-oriented simulation. Performance predictions are promising (see Section [7](#)). A number of less obvious advantages show themselves in a more detailed investigation. For example, it appears that these systems organically form and decompose virtual manufacturing cells without the need to pre-specify them [\(9\)](#). (A virtual manufacturing cell is a small pre-specified sub-system of machines which are temporarily aggregated as a cell without physical collocation. The virtual cell is described by McLean et al. in [\[40\]](#))

There are many shortcomings in the system described here and it will not work in all circumstances. We believe, however, that advancing technology in CNC machine tools, in process planning and in computers is progressively making the "monolithic FMS" solution to flexible manufacturing system control unacceptably ineffective. Not only are these systems not flexible, but they do not take advantage of the technologies which are becoming available.

The system described is dynamic and flexible in the long-term and straightforward to reconfigure; new machines may be added and old ones removed without affecting the system as a whole. A less sophisticated degree of expertise is required of the operators in the system, and they are able to

interact with it without having to have an in-depth understanding of any broad systemic implications of their local actions. While the shop takes advantage of those tasks which humans are good at, it is much less chaotic and controlled than a manual shop. For example, as the plant learns about successful shop-floor procedures, they become embodied in the controllers of the entities in the system, rather than being forgotten as they might be in a manual shop. Failure and recovery are handled by virtue of the structure of the system rather than by anticipating actions for every possible failure state in a central computer which won't know what the failure state is anyway.

There are some useful analogies from economics (this is like a market) or from organizational behavior (this is like cooperation/negotiation). While these things may be true, and may provide some insights, it is important to emphasize that this is a straightforward *engineering* solution to the problem of distributed orchestration, and came from application of engineering and computer-science techniques rather than from a forced economic or anthropomorphic analogy. Clearly, societal manufacturing systems like this are a few years away. However, we are convinced that such structures provide a clear direction for advance in manufacturing systems, particularly in machining. While they may be "sub-optimal" in the short-term Operations Research sense, in the long-term, they will provide long-lasting, robust flexibility and avoid the static dead end which FMS have progressively become.

7. Experimental Results

The system described above was, in its various aspects, simulated in SLAM II and G2. This model was developed over the period 1986-1991 and has now been adapted and enlarged by researchers at Purdue's Center for Intelligent Manufacturing Systems, Loughborough, Cambridge and Harvard. The system modelled comprised between 10 and 50 different machines in a heterarchical structure. The *maximal* system is shown diagrammatically in [Figure 13](#). The communications systems was modelled as a limited resource, along with the automated vehicle system. Monte Carlo experiments were carried out after validation, using an exponential demand stream for products which had been generated from expert estimates of processing time distribution parameters. Various failures and operating conditions were imposed on the system. In order to reproduce the experiments, readers are invited to email the author for the code. (A discussion of more general queueing network models for the scheme, and a comparison of simulation results with such queue-theoretic models is provided by Upton, Barash and Matheson in [\[41\]](#))

7.1 Utilization

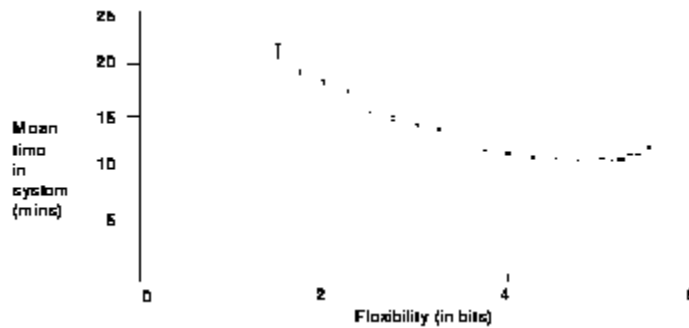
As one might expect, system utilization increases as jobs arrive to be processed more frequently. When the system is very lightly loaded, the faster machines tend to process all of the jobs arriving. As the system becomes more heavily loaded, less effective machines gradually take on more work, until a saturation point is reached, when machines receive jobs with probability in proportion to mean processing rate across all jobs. At saturation, each machine had statistically identical utilization of around 85%. Each machine had local rules concerning the amount of work-in-progress it was allowed to attract to itself. In general, this was limited to two parts. This, of course, means that WIP is limited by a kanban type system. For this reason, WIP did not exceed 200 jobs, except in the case of system-wide communications failure. Lead times (throughput times) were between 2 and 3 times the total machine processing times for the product, depending on the loading of the system. More details are provided in [\[41\]](#).

8. Performance and Routing Flexibility

A key issue which the experiment was designed to explore was the following: How does the flexibility of the machines affect the applicability of a heterarchical structure. Routing flexibility here is the

ability of the various machines to process multiple jobs[18],[42]. An entropic measure of machine flexibility was developed for the purpose of this experiment and is described in Section 9. As flexibility increases, the performance of the system improves, as one might expect. However, the system performance deteriorates as machines become very flexible, when they are able to work on almost every job. This is a result of the fact that the number of bids made for each job tends to jam the communications system and generally adds to the complexity of a job's decision about which machine to assign itself. Surprisingly, it became clear that there were some advantage for the heterarchy in specializing some machines slightly. Figure 8 shows some results from these experiments. Processing times are short in this experiment in order to stress the control system.

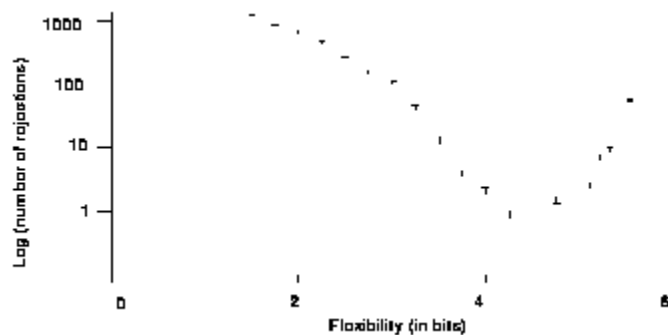
Figure 8 Mean Time in System versus Flexibility (200 jobs)



This phenomenon can be better understood by considering the rejection results in [Figure 9](#). When a job can find no takers (i.e. no machine bids because buffers are full) it waits for a short period of time before re-submitting itself to auction. If it still finds no takers, it extends its waiting period. This "exponential back-off" avoids communications saturation when the system becomes very busy.

When machines bid, they do so by taking into account bids they have already made, and keep a "pending list" of jobs they might win. This conservative approach avoids the possibility that they might win too many jobs and have their buffer overflow. If a machine does not hear of the outcome of the auction, the bid expires and it forgets it. When nearly all machines can process all part types, most machines tend to have long pending lists, which means they are less likely to bid. At the same time, the probability of any particular machine winning is reduced, so pending lists guard against increasingly improbable events. This causes the problem of very flexible systems rejecting jobs. Of course, in general, increases in flexibility improve the performance of the system.

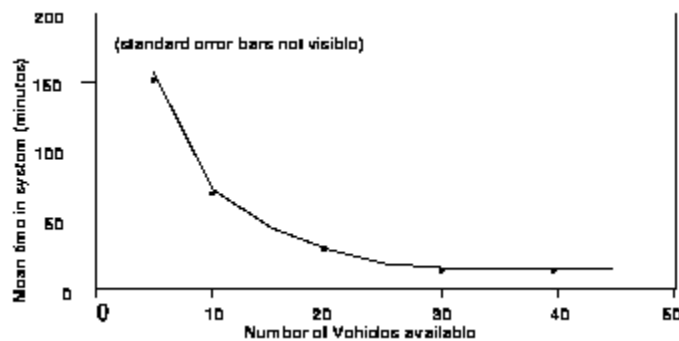
Figure 9 Number of Rejections versus Flexibility (200 jobs)



8.1 Tolerance of Failures

Machine failures had no effect on the system apart from the loss of capacity of the individual machine, and a tendency for longer throughput times by those parts in the failed machines' buffers at the time of failure. The system was made tolerant of communications failures by ensuring that all of the entities had sensible rules, which would avoid their "hanging" while waiting for a message. All entities used time-outs or similar alternatives to avoid this. Local communications failures and short, system wide communications failures had little effect on the system, but long, system-wide failures (more than a few minutes) caused failures which would demand some human intervention, parts arriving at a machine unannounced, for example, causing a buffer overflow. It should be emphasized that these failures occurred under fairly severe conditions, and also serve to underscore the fidelity of the simulation. Part failures were not explored and certainly should be. A part failure would manifest itself in some indirect way by the physical obstruction of other jobs or by failing to respond to queries from the supervisory computer. Vehicle failures did not cause system wide problems, except when unreasonably severe (see below).

Figure 10 Mean time in system versus Vehicles available



8.2 Resource requirements

The system was clearly constrained by lack of vehicles if there were fewer than 30 vehicles in the system. The communications resource is critical. The exact requirements depend heavily on the amount of data passed in the bid request messages). It is clear, however, that any problems in the ability to put packets on the network cause fairly severe problems, consistent with those caused by communications failure.

8.3 Further considerations

8.3.1 Product limitations

This system is designed to manufacture the types of products made by current FMS. While the ideas may be extendable to other products and manufacturing systems in general it is suitable for machined metal products produced in very small batches. The processing machines which make up the system must, in general, be autonomously computer controlled as they usually are in an FMS.

8.3.2 Process decomposition

There must be a clear decomposition of the processes which are to be performed on a product. In the case of, say, machined aircraft parts, jobs are often fixtured once, all accessible faces are machined by one machine then the part is turned over and re-fixtured to be machined on the other side. This would be an ideal application. If machines are allowed to perform partial operations between re-fixturing, and to bid for a partial job, then the part's bid evaluation procedure becomes much more

complex. How should a part decide on the utility of having only part of a job performed? It could negotiate ahead in its process until it has formed a complete path for itself, but the process becomes messy. This is definitely an avenue for further research.

8.3.3 Sequentiality

If a job requires processing by a large number of machines in sequence, the parts will make decisions which are too myopic. They may opt for a machine which is the most appropriate in the short-term, but find themselves a long way from their optimal total processing path. This is the justification often given for centralized control - that there must be some entity with a good temporal overview. Again, the system could be extended to cover such systems with more advanced negotiation schemes. Each move away from simplicity however, makes the argument for this type of system weaker. The strength of the method relies on the appropriate application of the method.

9. Acknowledgments

Acknowledgments are due to Professors Moshe Barash at Purdue, Ramchandran Jaikumar at Harvard and David Williams at Loughborough. The collaborative assistance of the Engineering Research Center for Intelligent Manufacturing Systems at Purdue University is gratefully acknowledged. The comments of the anonymous referees were most constructive.

References

- [1] Williamson, D. T. N
"System 24 - A New Concept of Manufacture". Proceedings of the 8th International Machine Tool and Design Conference. pp. 327-376 Pergamon Press, 1967.
- [2] General Electric
"Product /System productivity research." Report prepared by General Electric Company for the National Science Foundation, Washington D.C., 1976.
- [3] Barash, M M,
"Manufacturing Systems Today and Tomorrow," International Journal of Advanced Manufacturing Technology, vol. 1 (2), pp. 1-4, IFS (Publications) Ltd., 1986.
- [4] Barash, M M, F F Leimkuhler, C L Moodie, S Y Nof, J J Solberg, and J J Talavage,
"Optimal Planning of Computerised Manufacturing Systems (CMS)," NSF Grantees Conference, Cornell University, September 1979.
- [5] Hartley, John,
"FMS at Work", pp. 133-151, IFS (Publications) Ltd., Bedford, UK, 1984.
- [6] O'Grady, P. J. and U. Menon,
"A Concise Review of Flexible Manufacturing Systems and FMS Literature," Computers in Industry 7, pp. 155-167, Elsevier Science Publishers B. V., 1986.
- [7] Barbera, Anthony J., M. L. Fitzgerald, and J. S. Albus,
"Concepts for Real-Time Sensory-Interactive Control System Architecture," Proceedings. 14th Southeastern Symposium on System Theory, pp. 121-126, April, 1982.
- [8] McLean, Charles R.,
"Information Architecture of the Automated Manufacturing Research Facility (AMRF)," Proceedings of the Information Technologies Conference, Troy, NY., June, 1986.
- [9] McLean, Charles and A Jones,
"A Proposed Hierarchical Control Model for Manufacturing Systems," Journal of Manufacturing Systems, vol. 5 (1), January, 1986.
- [10] Jaikumar, Ramchandran,
"Postindustrial Manufacturing" The Harvard Business Review, November/December 1986.
- [11] Bolwijn, P. T. and T. Kumpe.
"Towards the Factory of the Future". The McKinsey Quarterly, Spring 1986.
- [12] Ingersoll Engineers.

- The Ingersoll FMS Report, 1982. Ingersoll Engineers, Rugby, UK.
- [13] Tidd, Joseph
"Flexible Manufacturing Technologies and International Competitiveness". Pinter Publishers, 1991.
- [14] Ranta, Jukka and Iouri Tchijov
"Economics and Success Factors of Flexible Manufacturing Systems: The Conventional Explanation Revisited" The International Journal of Flexible Manufacturing Systems 2 (1990) pp. 169-190.
- [15] Kelley, Maryellen R.
"The State of Computerized Automation in US Manufacturing". Harvard University John F. Kennedy School of Government, 1988.
- [16] Kearney and Trecker Marwin Ltd.,
Sales Literature, KTM Ltd., Brighton, England, 1985.
- [17] Medearis, H. D. IV, M M Helms and LP Ettkin.
"Justifying flexible manufacturing systems from a strategic perspective". Manufacturing Review 3 (4). December 1990.
- [18] Buzacott, J A,
"The Fundamental Principles of Flexibility in Manufacturing," Proceedings. 1st International Conference on FMS, pp. 13-22, IFS (Publications) Ltd., Bedford, U.K., 1982.
- [19] Bjørke, O.,
"Software Production - The Bottleneck of Future Manufacturing Systems," Annals of the CIRP, vol. 4 no. 2, pp. 545-548, 1975.
- [20] Hutchinson, G. K. and A. T. Clementson,
"Manufacturing Control Systems: An Approach to Reducing Software Costs," Robotics and Computer Integrated Manufacturing, vol. 1 No. 3/4, pp. 271-281, Pergamon, 1984.
- [21] Jaikumar, Ramchandran,
"FMS - A Managerial Perspective". Harvard Business School working paper. January 1984.
- [22] Bereiter, S.R., and Miller, S.M., (1989),
"A field based study of troubleshooting in computer-controlled manufacturing systems". IEEE Transactions on Systems Man and Cybernetics, smc-19, 205-219.
- [23] Fox, B R and K G Kempf,
"A Representation for Opportunistic Scheduling," IEEE Conference. on Robotics and Automation, pp. 111-117, 1985.
- [24] Hatvany, Jozsef and Jozsef Janos,
"Software Products for Manufacturing Design and Control," Proceedings of the IEEE, vol. 68 No. 9, pp. 1050-1053, September, 1980.
- [25] Lewis, W., M. M. Barash, and J. J. Solberg,
"Computer Integrated Manufacturing System Control: A Data Flow Approach," Journal of Manufacturing Systems, vol. 6, no. 3, pp. 177-191.
- [26] Lewis, William C.,
"A Data Flow Architecture for Manufacturing System Control" Ph.D. Dissertation, Purdue University, West Lafayette, IN, 1984.
- [27] Maley, James G. and James J. Solberg,
"Part Flow Orchestration in CIM," 9th International Conference on Production Research, Cincinnati, August 17-20, 1987.
- [28] Parunak, H. Van Dyke, Bruce W. Irish, James Kindrick, and Peter W. Lozo,
"Fractal Actors for Distributed Manufacturing Control," The Second Conference on Artificial Intelligence Applications, pp. 653-660, IEEE, 1985.
- [29] Vamos, T.,
"Cooperative Systems based on Non-Cooperative People," IEEE Control Systems Magazine, vol. 3, pp. 9-14, 1983.
- [30] Agha, A
"Actors: A Model of Concurrent Computation in Distributed Systems", MIT Press, 1986.
- [31] Smith, Reid G.,
"The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver," IEEE Transactions on Computers, vol. C-29 No. 12, pp. 104-113, 1980.
- [32] Duffie, Neil A. and Rex S. Piper,

- "Nonhierarchical Control of Manufacturing Systems," *Journal of Manufacturing Systems*, vol. 5 No.2, pp. 137-139, 1986.
- [33] Parunak, H. Van Dyke,
"Manufacturing Experience with the Contract Net," in *Distributed Artificial Intelligence*, Michael M Huhns, pp. 285-310, Pitman, London, 1987.
- [34] Shaw, Michael J.,
"Task Bidding and Distributed Planning in Flexible Manufacturing," *Proceedings of the 2nd Conference on Artificial Intelligence Applications*, pp. 184-189, IEEE, Miami Beach, Florida, 1985.
- [35] Dilts, D. M., Boyd, N. P., Whorms, H. H.
"The Evolution of Control Architectures for Automated Control Systems". *Journal of Manufacturing Systems*. Vol. 10 (1) pp. 79-63. 1991.
- [36] DeGarmo, E. Paul, Black, J. and Kohser, R. A.
"Materials and Processes in Manufacturing", 7th Edition, MacMacmillan. pp. 760-763
- [37] Anonymous Reviewer for
"Manufacturing Review". 1991.
- [38] Yange, E-H.
Purdue University Engineering Research Center, Preliminary PhD Report, 1990. School of Industrial Engineering, West Lafayette IN 47907.
- [39] Veeramani, D.,
"Physical Resource Management in Large Computer-Controlled Manufacturing Systems", Purdue University, West Lafayette, IN, November, 1989.
- [40] McLean, C. R., H. M. Bloom, and T. H. Hopp,
"The Virtual Manufacturing Cell," National Bureau of Standards, pp. 1-9, Industrial Systems Division, Washington D. C., 1983.
- [41] Upton, D. M., M. M. Barash and A. M. Matheson
"Architectures and Auctions in Manufacturing Systems". *International Journal of Computer Integrated Manufacturing*, 1991.
- [42] Dapiran, A. and M. Manieri,
"The Cost of Flexibility in the FMS," *Proceedings of the Second International Conference on FMS*, pp. 555-568, IFS (Publications) Ltd., 1983.
- [43] Aneke, N A G and A S Carrie
"A Comprehensive Flowline Classification Scheme," *Int J. Prod. Res.*, vol. 22 No. 2, pp. 281-297, 1984.
- [44] Upton, D. M. and M. M. Barash,
"A Grammatical Approach to Routing Flexibility in Large Manufacturing Systems," *Journal of Manufacturing Systems*, vol. 7 (3), pp. 209-221, 1988.
- [45] Upton, D.M. and M. M. Barash,
"The Generation of Random Families of Parts for Manufacturing Systems Simulation". *International Journal of Computer Integrated Manufacturing*, 2 (3), 1989.
- [46] Schmeiser, B. W. and Lal, R.
"A Five-parameter Family of Probability Distributions." Technical Report 84-7. School of Industrial Engineering, Purdue University. West Lafayette, IN 47907. April 1984
- [47] Yao, D. D.
"Material and Information flows in Flexible Manufacturing Systems", *Material Flow*, V 2, 143-149, 1985
- [48] Kumar, V.
"On measurement of Flexibility in Flexible Manufacturing Systems: An information-theoretic approach." *Proceedings. 2nd ORSA/TIMS Conference on FMS: Operations Research Models and Applications*. pp. 131-143, Elsevier Science Publishers B.V., Amsterdam, 1986

APPENDIX: The Generation of Random Manufacturing Systems

To explore the performance of the architecture over a range manufacturing systems with varying routing flexibility, it is necessary to generate "random" manufacturing systems. In general, this is not straightforward. The key feature of the manufacturing system which we were interested in exploring here was its short-term routing flexibility. Manufacturing systems here provide a job with high routing flexibility when

- The expected processing times on each of the feasible machines is similar, and
- The number of machines which may perform the job is high.

We are interested in generating a matrix of mean processing times for the jobs in the product range. Some of these processing times will be 'infinite', since machines may be unable to perform some tasks. Once this matrix has been generated, the routing flexibility of the manufacturing system has been characterized and may be measured as described in Section A2.1. A fixed transfer line has no routing flexibility at a given process step [43] while a multi-machine job shop has high routing flexibility. The system we describe here is one in which machines perform one set of operations on a job, then the job returns for re-fixturing. Thus, jobs here may also be new operations sets on partially machined parts. It is straightforward to extend the methods shown here to systems in which parts go from one machine to the next, since the output buffer of the previous machine may be considered to be the new input buffer for that job. Extensions to highly sequential systems are discussed in [44].

A1. Generating the processing time matrix

In order to generate samples from a random matrix, \mathbf{T} , conforming to the description above, the following procedure is carried out. Samples of a matrix, \mathbf{H} , of independent, $N(0,1)$ random variates is generated, of dimension m rows (jobs) by n columns (machines). Each row and column vector of \mathbf{H} thus has covariance matrix \mathbf{I} . Let the desired correlation matrix of the row vectors of \mathbf{T} be \mathbf{R}_p and that of the column vectors be \mathbf{R}_m . To provide the necessary correlation structure between elements in a row, the column vectors are transformed by linear transformation such that

$$(1) \quad \mathbf{H}' = \Phi \Lambda^{\frac{1}{2}} \mathbf{H}^T$$

where Φ, Λ are defined by

$$(2) \quad \mathbf{R}_m \Phi = \Phi \Lambda$$

$$(3) \quad \Phi^T \Phi = \mathbf{I}$$

Thus, Φ is an $n \times n$ matrix comprising the n eigenvectors of \mathbf{R}_m ,

$$(4) \quad \Phi = [\Phi_1 \Phi_2 \dots \Phi_n]$$

and Λ is a diagonal matrix of associated eigenvalues.

The matrix \mathbf{H}' is subjected to a further transformation such that

$$(5) \quad \mathbf{H}'' = \Psi \Xi^{\frac{1}{2}} (\mathbf{H}')^T$$

$$(6) \quad \mathbf{R}_p \Psi = \Psi \Xi$$

$$(7) \quad \Psi^T \Psi = I$$

So again, Ψ is an $m \times m$ matrix comprising the m eigenvectors of R_p ,

$$(8) \quad \Psi = [\Psi_1 \Psi_2 \dots \Psi_m]$$

and Ξ is a diagonals matrix of eigenvalues.

This matrix will induce exact inter-column correlations in normal variates but will, in general, disrupt the correlations induced within the rows by (1). It may be shown, however, that providing the correlation of jobs' processing times on different machines is high, the corruption of the inter-machine correlation structure after inducing the inter-job correlation structure by (5) is negligible. These errors are characterized in [45]. Marginal distributions are generated from the correlated standard normals using the procedure outlined by Schmeiser and Lal [46].

A sample mean processing time matrix is shown in [Figure 11](#).

A2. Generating the Target Flexibility

A2.1 Measurement of Routing Flexibility

The measure used for flexibility here is an entropic measure borrowed from information theory. Such measures are discussed in the literature in [47], [44] and [48]. The brief justification for such a measure (details are in the appropriate papers) is that a measure of routing flexibility should:

- Rise smoothly with the number of candidate processing stations.
- Rise smoothly with the indifference among the various options.

Let the processing rate of job i on machine j be $\tau_{ij} = T_{ij}^{-1}$. We may define a vector, π_{ij} , such that

$$(9) \quad \pi_{ij} = \frac{T_{ij}^{-1}}{\sum_j T_{ij}^{-1}}$$

This means that (for example) a job for which there is only one processing machine of the n in the system has $\pi = (1, 0, \dots, 0)$ and one which may be processed by all machines at equal rates has

$\pi = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})$. We define the mean flexibility afforded by the system to part i to be given by

$$(10) \quad \Phi_i = - \sum_{j=1}^n \pi_{ij} \log_2 \pi_{ij}$$

And the expectation of flexibility across parts as:

$$(11) \quad \xi \bar{\Phi} = \xi [\Phi_1, \Phi_2, \dots]$$

where ξ_i is the proportion of demand for job i . The probability of a job being of a particular type is an exogenous variable, so the entropy associated with this choice is not included in the measure of the system flexibility. If all part types are equally in demand, the flexibility of a system with the matrix shown in [Figure 11](#) is 5.56 bits. If parts were routed to machines with probability in proportion to their processing rates (which they are at high loads, see [41]), this measure tells us the

theoretical minimum amount of information a central control computer would have to transmit to the system in order to route the part. It thus gives us a measure of the *complexity* of the routing decision. For n machines this is a maximum of $\log_2(n)$ bits(10) and a minimum of 0 bits.

A2.2 Restricting the choice set to reduce flexibility

A manufacturing system in which every machine can produce every job more or less quickly, has a maximum flexibility quantified by (10). Of course, in general, it is the inability of a manufacturing machines to perform all jobs that inhibits flexibility. To reduce the flexibility of the manufacturing system, machines are randomly deprived of the ability to perform various jobs until the target flexibility is reached. Machines without the ability to perform a job, will of course not bid on it. This allows us to create systems which more closely reflect reality - machines are usually specialized to some extent. It also allows us to reduce the system size by removing the ability of some machines to produce any jobs. The procedure is as shown in Figure 12. The target routing flexibility Φ is entered, along with an error tolerance, ϵ from within the simulation model. A manufacturing system is generated by the procedure and then used by the simulation model as the basis for that particular run of the experiment.

Figure 11 Sample of Processing Time Matrix for Experimental Part Family

	Part Type									
	1	2	3	4	5	6	7	8	9	10
Machine 1	8.11	8.17	13.47	8.86	5.48	14.80	8.80	8.52	10.73	7.27
Machine 2	5.88	5.61	5.86	5.22	5.06	16.67	6.10	5.77	6.78	5.33
Machine 3	5.31	5.58	5.38	5.10	5.03	8.06	8.36	5.81	5.56	5.14
Machine 4	5.10	5.18	5.28	5.10	5.03	8.26	5.41	5.18	6.36	5.13
Machine 5	7.11	6.57	6.31	6.58	5.73	18.17	11.50	8.66	12.18	8.61
Machine 6	5.07	5.04	5.16	5.06	5.03	6.32	5.50	5.08	5.28	5.07
Machine 7	21.71	17.62	16.36	15.10	7.74	44.87	28.78	24.38	25.53	10.01
Machine 8	11.84	10.40	10.70	7.76	6.28	22.64	23.66	14.86	18.78	8.00
Machine 9	15.88	10.61	14.18	7.18	6.24	31.06	13.26	16.38	28.24	8.40
Machine 10	8.18	5.88	8.66	6.38	5.16	23.86	8.84	7.76	13.60	7.42
Machine 11	5.26	5.06	5.08	5.01	5.01	6.32	5.31	5.14	5.08	5.08
Machine 12	7.58	5.63	5.22	5.38	5.06	8.62	7.17	5.80	5.88	5.52
Machine 13	5.04	5.08	5.20	5.11	5.03	6.33	5.54	5.06	5.62	5.11
Machine 14	5.77	5.23	6.13	5.36	5.04	11.06	7.16	6.31	8.62	6.76
Machine 15	5.63	6.33	6.62	5.52	5.08	11.73	6.33	5.62	7.06	5.14
Machine 16	5.76	6.14	6.18	5.32	5.07	8.81	5.82	5.78	12.76	5.68
Machine 17	5.61	5.74	5.68	5.18	5.08	8.60	6.18	6.26	8.08	5.82
Machine 18	5.81	5.08	5.26	5.04	5.04	6.51	5.54	5.06	6.07	5.28
Machine 19	28.71	15.12	16.21	7.48	8.88	51.30	21.42	28.08	27.66	27.03
Machine 20	5.06	5.06	5.08	5.02	5.06	6.30	5.33	5.32	5.81	5.13
Machine 21	6.13	6.03	6.66	5.66	6.07	16.28	8.36	8.63	10.06	8.66
Machine 22	5.08	5.00	5.11	5.02	5.01	5.36	5.13	5.07	5.02	5.06
Machine 23	8.16	6.47	16.53	7.76	5.33	12.48	21.28	8.47	21.41	6.86
Machine 24	6.04	6.82	7.33	5.58	5.88	10.82	8.61	5.52	8.31	5.38
Machine 25	8.17	11.16	14.88	7.81	6.58	26.12	15.66	21.47	13.64	10.02
Machine 26	6.44	7.16	6.46	5.54	5.82	28.62	8.74	11.78	8.44	7.50
Machine 27	25.06	10.18	18.62	17.34	6.84	35.43	26.07	15.86	34.67	12.26
Machine 28	8.44	10.01	7.73	6.74	6.68	34.10	14.68	10.04	18.43	7.88
Machine 29	6.36	6.68	7.27	5.28	5.68	18.88	6.84	7.02	8.56	6.32
Machine 30	10.01	16.27	8.57	8.16	10.12	38.84	18.28	8.86	17.37	17.13
Machine 31	5.03	5.01	5.28	5.07	5.06	5.76	5.62	5.11	5.33	5.12
Machine 32	5.16	5.06	5.30	5.01	5.06	6.30	5.61	5.16	5.48	5.32
Machine 33	6.83	6.26	6.77	5.33	5.66	13.86	8.06	6.48	12.66	6.18
Machine 34	10.03	8.04	15.51	7.21	7.50	35.38	15.70	10.86	8.88	8.16
Machine 35	5.02	5.01	5.06	5.01	5.02	6.01	5.28	5.06	5.13	5.00
Machine 36	14.57	18.17	17.38	11.87	7.74	41.28	36.88	18.28	25.46	10.77
Machine 37	5.26	6.16	6.24	5.46	5.26	10.58	6.88	5.88	5.73	5.44
Machine 38	15.63	11.03	21.28	8.48	6.61	35.80	22.88	8.06	22.71	10.41
Machine 39	13.63	8.56	10.47	5.78	6.78	27.10	18.88	10.08	8.13	7.42
Machine 40	8.46	7.80	10.58	8.08	5.82	22.16	11.86	6.33	16.47	5.66
Machine 41	5.83	5.40	5.82	5.27	5.18	11.12	8.18	5.86	7.73	5.50
Machine 42	5.22	5.16	5.32	5.04	5.06	6.27	5.71	5.16	5.53	5.06
Machine 43	5.00	5.00	5.00	5.00	5.00	5.06	5.00	5.00	5.00	5.00
Machine 44	8.76	8.30	11.30	6.74	7.48	32.68	20.56	8.20	25.00	8.88
Machine 45	6.31	5.81	5.31	6.61	5.44	11.58	6.02	5.36	7.04	5.88
Machine 46	7.64	7.81	10.37	7.03	5.74	12.52	13.11	7.54	11.46	6.42
Machine 47	5.70	5.88	5.28	5.08	5.06	15.08	6.52	5.54	6.48	5.36
Machine 48	13.26	6.38	8.10	5.72	6.36	20.47	8.86	10.86	11.52	5.80
Machine 49	6.10	6.10	10.68	5.87	5.16	10.67	7.88	6.67	6.78	6.87
Machine 50	8.28	7.76	11.26	5.33	7.18	18.18	11.04	7.34	14.26	12.16

(Note that this matrix describes processing times before imposing an incidence matrix to reduce flexibility as shown in Figure 12)

Figure 12 Procedure For Generating Manufacturing Systems With Variable Routing Flexibility

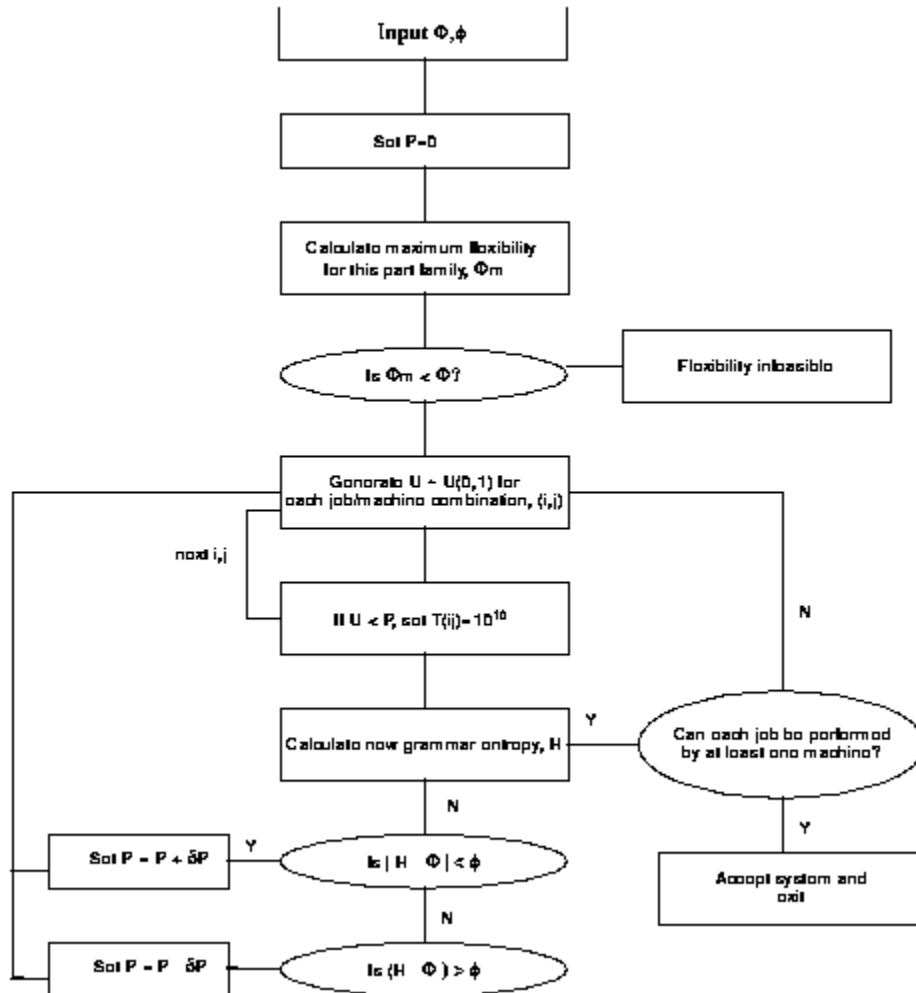


Figure 13 Schematic of Maximal System



David M. Upton is an Assistant Professor of Business Administration at Harvard University, where he teaches the second-year MBA course in Operations Strategy, as well as short courses in manufacturing for practitioners. He is a Chartered Mechanical Engineer, and has degrees in Manufacturing Engineering from Cambridge University and Industrial Engineering from Purdue University. His current research interest are in Computer Integrated Manufacturing and the Management of Flexibility in manufacturing systems.

List of Figures

[Figure 1 The Best of Both Worlds \(on page 2\)](#)

[Figure 2 A Flexible Manufacturing System \(on page 3\)](#)

[Figure 3 Hierarchical Control \(on page 4\)](#)

[Figure 4 Part transmits processing request and data to the system. \(on page 9\)](#)

[Figure 5 Machines transmit an estimate to the job. In this case, machine 3 is down, and fails to respond to the request. \(on page 12\)](#)

[Figure 6 The Collective Memory \(on page 14\)](#)

[Figure 7 Hybrid Control \(heterarchy within a hierarchy\) \(on page 16\)](#)

[Figure 8 Mean Time in System versus Flexibility \(200 jobs\) \(on page 21\)](#)

[Figure 9 Number of Rejections versus Flexibility \(200 jobs\) \(on page 22\)](#)

[Figure 10 Mean time in system versus Vehicles available \(on page 22\)](#)

[Figure 11 Sample of Processing Time Matrix for Experimental Part Family \(on page 30\)](#)

[Figure 12 Procedure For Generating Manufacturing Systems With Variable Routing Flexibility \(on page 31\)](#)

[Figure 13 Schematic of Maximal System \(on page 32\)](#)

Footnotes

(1)

Adapted from "The Economist", May 30, 1987.

(2)

(now National Institute of Standards and Technology)

(3)

Some systems companies, such as KTM, advocated a piecemeal approach to FMS installation to allay this fear. First the machine tools would be installed, then the materials handling system and then the central control computer. The company could learn about each new technology (and generate funds with it) before progressing to the next stage [16].

(4)

I am grateful to an anonymous reviewer for his or her additional hypotheses in this section

(5)

This quality is often called short-term flexibility in academia - see [18]. The ability to change the system to produce new products is "long-term" flexibility.

(6)

IIASA is the International Institute for Applied Systems Analysis in Vienna. See [14].

(7)

The part need not retain the experiences of every previous part. There can simply be tables of estimates like "expected performance", which a part may update from its own individual experiences in the system, with some appropriate weighting.

(8)

This is simply exponential smoothing, a standard forecasting technique which may be found in any textbook on the subject.

(9)

The fact that tooling is already available at a machine means that it consistently wins auctions for parts belonging to a similar family, while there is a demand. The next machine in the process also tends to accumulate tooling for a prevailing family, hence a virtual cell is formed and tooling tends to stay in one place. As demand disappears, the cell fades away.

(10)

= 5.644 for n= 50