

# Self-Organizing Manufacturing Control: An Industrial Application of Agent Technology\*

Stefan Bussmann

Klaus Schild

DaimlerChrysler AG

Research & Technology 3

Alt-Moabit 96A, 10559 Berlin, Germany

E-mail: {stefan.bussmann,klaus.schild}@daimlerchrysler.com

## Abstract

*We present an auction-based approach to manufacturing control. Workpieces auction off their current task, while machines bid for tasks. When awarding a machine, a workpiece takes into account not only the machine's current work in process, but also the outgoing flow of material. If a machine's outgoing stream is blocked, eventually the machine will not accept a new workpiece, thus blocking its input stream as well. As a result a capacity bottleneck is automatically propagated in the opposite direction of the material flow. A unique feature of this mechanism is that it does not pre-suppose any specific material flow; the current capacity bottleneck is always propagated in the opposite direction of the actual flow, no matter what this flow looks like. This paper includes a detailed analysis of the mechanism, including a formal proof of its freedom of deadlocks.*

*DaimlerChrysler evaluated the new control approach as a bypass to an existing manufacturing line. A suite of performance tests demonstrated the industrial feasibility and the benefits of the approach.*

## 1. Introduction

Today's manufacturing industry is facing a major shift from a supplier's to a customer's market. The growing surplus of industrial capacity provides the customer with a greater choice and increases the competition between suppliers. Aware of their power, customers have become more demanding and less loyal to a particular product brand. As a result, companies must shorten product-life cycles, reduce

time-to-market, increase product variety and instantly satisfy demand, while maintaining quality and reducing investment costs. This is a great challenge to the manufacturing process itself; it must be more flexible and robust as well as demonstrate enhanced scalability [1].

Agents are the right information technology to meet this challenge. They model the manufacturing process as it is with no artificial central control unit. Resources are allocated dynamically by a continuous coordination process among the relevant agents. Unlike in Computer Integrated Manufacturing (or CIM), there is no need to handle all of the contingencies of a complex manufacturing process at design time; rather, agents negotiate proper allocations among themselves during execution. Although some of their collective decisions may not be optimal, all decisions are, nevertheless, made on the basis of the actual situation—in the long run leading to a higher throughput.

Under the leadership of DaimlerChrysler, an industrial consortium was formed to meet the challenges of manufacturing in modern automotive industry. The consortium developed an agent-based manufacturing system. While the technical details of implementation are proprietary to the consortium, we will present the overall architecture of the system in this paper. We will also give a detailed analysis of the agent-based control mechanism, including a formal proof of its freedom of deadlocks. The analysis is complemented by a brief report on a series of realistic simulations as well as performance tests on an industrial prototype.

## 2. A Flexible & Robust Manufacturing System

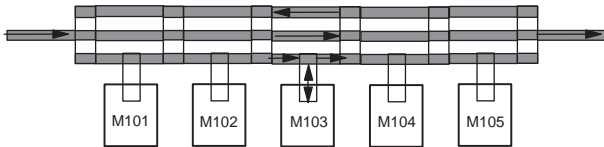
In a robust manufacturing system, machines must have overlapping capacities. This means that for every manufacturing step there is always more than one machine which, at

---

\*published at 4th Int. Conf. on Multi-Agent Systems, 2000

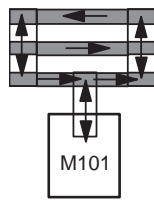
least in principle, is able to perform this operation. In case of a machine breakdown, this type of redundancy provides the system with the flexibility of diverting a pallet to another machine. Diverting a pallet, however, is not possible without being able to bypass a machine; in particular, with a number of other pallets in the waiting queue of the machine to be bypassed. In general, it should be possible for a pallet to bypass even more than one machine.

DaimlerChrysler developed the concept of a modular manufacturing system that meets this criterion. Fig. 1 shows an example of the new layout. The entire manufacturing



**Figure 1. A Flexible Manufacturing System**

system is composed of standard modules. Each of these modules consists of a machine, three one-way conveyors, two switches and a shifting table. The seven components of a module are arranged as in Fig. 2. Every switch can move

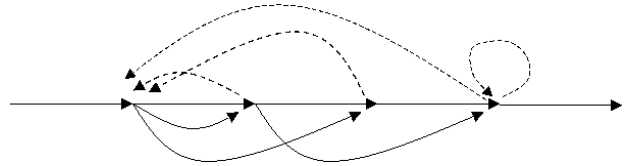


**Figure 2. Standard Module**

a pallet from any of its entries to any of its exits, in each case one pallet at a time. In Fig. 1, each of the intermediary switches has two entries and one exit at the left-hand side, while at the right-hand side there are two exits and only one entry.

An arrangement of standard modules as in Fig. 1 makes it possible for a pallet to either enter a machine through the lowermost conveyor or else bypass the machine through the middle conveyor. After having bypassed a machine through the middle conveyor, a pallet has two options: it can either proceed in a forward direction to a subsequent machine or move backwards using the topmost conveyor. If, for instance, the lowermost conveyor is already occupied, preventing a pallet from entering the target machine, then the pallet can move backwards and forwards in a circle until the lowermost conveyor is available again. In this way, the entire transportation system serves as a flexible buffer.

Most transportation systems induce specific constraints on the flow of material; that is, the sequences in which pallets may visit machines. Let  $M$  and  $M'$  be two machines.



**Figure 3. Admissible and Principal Flow**

If in the transport system there is a path from  $M$  to  $M'$ , then there may be a material flow from  $M$  to  $M'$ ; otherwise, a material flow from  $M$  to  $M'$  is impossible. A layout such as the one depicted in Fig. 1, however, does not impose any constraint whatsoever on the material flow; between any pair of machines, there may be a material flow in either direction. For performance reasons, it is often convenient to impose some constraints anyway.

**Definition 1** For every manufacturing system there is a special pre-defined binary relation among machines, denoted by  $\prec$ . This relation defines the admissible ordering in which a workpiece may visit machines:  $M'$  is an admissible successor of  $M$  iff  $M \prec M'$ . A workpiece may move from a machine only to one of the admissible successors of that machine; no other successor is possible. The relation  $\prec$  is called the admissible flow of material.

The admissible flow of material, of course, has to comply with the constraints of the transportation system; however, not every flow that would be possible according to the transportation system alone must be included.

An admissible flow of material may well have cycles. In fact, repair cycles are actually quite common in manufacturing. However, cycles reduce the overall throughput and thus a backward loop should be avoided whenever possible.

**Definition 2** A principal flow of material,  $\prec_p$  in symbols, is an acyclic subset of  $\prec$ ; that is,  $\prec_p$  is a subset of  $\prec$  such that there is no machine  $M$  with  $M \prec_p^+ M$ , where  $\prec_p^+$  denotes the transitive closure of  $\prec_p$ .

This means that the admissible flow  $\prec$  is partitioned into two disjoint subsets, a major flow  $\prec_p$  and a minor flow  $\prec \setminus \prec_p$ . The motivation behind this distinction is that the major flow represents the main manufacturing direction. As it stands, Definition 2 is still too general to exactly capture this intuition. To see this, consider Fig. 3. Our intuition tells us that the solid lines denote the principal flow, whereas the minor flow is depicted by the dashed lines. Definition 2, however, allows for other interpretations as well. According to Definition 2, the minor flow could just include the cycle with length one and the forward edges in the middle. The principal flow would then cover every other edge, nevertheless being acyclic.

Such unintended interpretations can be easily eliminated. We just have to require that a minor flow from  $M_n$  to  $M_0$

with  $M_n \neq M_0$  must not only be part of the admissible flow, but it also must run in the opposite direction of a path  $M_0, \dots, M_n$  in the major flow. Mathematically, this translates into the following additional constraint:

$$\prec \setminus \prec_p \subseteq (\prec_p^*)^{-1}. \quad (1)$$

This states that the minor flow  $\prec \setminus \prec_p$  is always a subset of the inverse ( $^{-1}$ ) of the reflexive and transitive closure ( $^*$ ) of the principal flow  $\prec_p$ . Throughout this paper, we assume that this additional condition is always met. With this restriction it in fact makes sense to distinguish between forward and backward successors.

**Definition 3** Let  $M$  and  $M'$  be machines.  $M'$  is called a forward successor of  $M$  iff  $M \prec_p M'$ , while  $M'$  is a backward successor of  $M$  iff  $M'$  is an admissible successor but not a forward successor of  $M$ .

### 3. Self-Organizing Control of Material Flow

To control the flexible manufacturing system presented in the previous section, we have developed a strictly decentralized approach to manufacturing control, called *West*.<sup>1</sup> In this approach, a specific agent is associated with each workpiece, each machine, and each switch. A *workpiece agent* manages the state of the workpiece attached to a specific pallet. A *machine agent* controls the overall material flow through a machine, not just the work in process. To this end, every machine agent manages what we call a *virtual buffer*. This buffer includes not only the machine's current work in process, but also the outgoing flow of material; that is, all those workpieces which have already been processed by the machine without yet being able to find an appropriate new machine. A third type of agent, a *switch agent*, controls a particular switch. It decides which entry to serve first and where to move a pallet.

All these agents constitute parallel processes. These processes are, of course, not independent; they have to be coordinated. Proper coordination is achieved by special negotiation procedures, which also take place simultaneously. A single workpiece negotiates with the machines about which of the machines should process the workpiece next.<sup>2</sup> The workpiece auctions off its current due operations; it invites machines to bid. Every machine bid includes information about the current state of the machine's virtual buffer. If a workpiece awards a specific machine, then this machine will be the next goal of the workpiece. The routing of a workpiece is organized through a sequence of bilateral negotiations, in each case between the workpiece and the next

<sup>1</sup> *West* is an abbreviation for the German word *Werkstücksteuerung*.

<sup>2</sup> Whenever understood, we ignore the distinction between an agent and the physical component it controls.

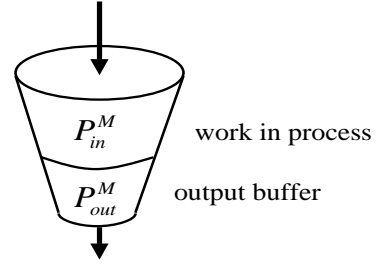


Figure 4. Virtual Buffer

switch which the workpiece approaches until the workpiece eventually has reached its next goal.

This is the *West* approach in a nutshell. The details are elaborated in the following subsections.

#### 3.1. Controlling the Flow through a Machine

Each machine agent manages two buffers, an *input* and an *output buffer*. The input buffer contains all those workpieces which awarded the machine and have not been processed yet. This is the machine's work in process. The number of workpieces in the input buffer of machine  $M$  is denoted by  $P_{in}^M$ .

A machine's output buffer tracks all those workpieces that already have been processed by the machine without yet being able to award an appropriate new machine. A workpiece thus moves from the input to the output buffer after being processed by the machine. The number of workpieces in  $M$ 's output buffer is denoted by  $P_{out}^M$ . The input and output buffer together constitute what we call a *virtual buffer*. Fig. 4 illustrates the structure of a virtual buffer.

The size of  $M$ 's virtual buffer is  $P^M = P_{in}^M + P_{out}^M$ .  $P^M$  is always bounded above by a specific constant  $P_{max}^M \in \mathbb{N}$ . This constant may vary from machine to machine. It should, however, never exceed the actual capacity of the physical buffer associated with the machine; that is, the section of the transportation system located between two neighboring switches.

**Definition 4** Let  $P^M = P_{in}^M + P_{out}^M$  be the current size of the virtual buffer of a machine  $M$ . Then  $M$  is ready to accept a new workpiece iff  $P^M < P_{max}^M$ .

In the presence of more than one product type, this condition is no longer sufficient. Multiple product types can be handled by additionally tracking the specific types of the workpieces in the output buffer and appropriately modifying the above definition. Details, however, cannot be provided within the space limitations of this paper.

### 3.2. Dynamic Task Allocation

Each workpiece agent manages the state of a specific workpiece. To this end, the agent has to be aware of the particular type of the workpiece. A *product type* is characterized by the exact sequence of operations which, one after the other, are to be applied to the workpiece. The *state* of the workpiece consists of the sequence of operations already performed.

**Definition 5** Let  $w$  be a workpiece with product type  $(o_1, \dots, o_n)$  and state  $(o_1, \dots, o_i)$ , where  $i \leq n$ . There remains a (possibly empty) sequence of operations to be done, that is,  $(o_{i+1}, \dots, o_n)$ . This sequence is called the current task (or the due operations) of  $w$ . If  $\vec{o} = (o_1, \dots, o_n)$  is an arbitrary sequence of operations, then every non-empty initial subsequence of  $\vec{o}$ , i.e., every sequence  $(o_1, \dots, o_i)$  with  $1 \leq i \leq n$ , is called a subtask of  $\vec{o}$ .

The current task (or at least a subtask) has to be allocated to a proper machine. In *West* this allocation is determined during execution. It would be impossible to make use of the flexibility provided by machines offering the same operation without such a late commitment. This dynamic allocation is carried out by a simple first-price, single-round auction, with which a workpiece auctions off its current task. More sophisticated auctions, like a Vickrey auction, are not needed here [9]. This is because all agents are, in principle, designed by one single authority in the domain of manufacturing control. Here, agents have only a very limited form of self-interest, thus avoiding problems of untruthful bidding. Providing an incentive to bid truthfully, however, is the main idea of the Vickrey auction.

The auction protocol used in *West* consists of only a single round. Each round involves three steps, the call for bids, the bidding itself, and the awarding. These steps are described in detail below.

Let us assume that  $op(M)$  denotes the set of operations which a machine  $M$  is able to perform in principle while ignoring short-term disturbances.

*Step 1.* The protocol is always initiated by a workpiece agent; in particular, whenever a workpiece first enters the manufacturing system and, thereafter, immediately after it leaves a machine. In any case, the workpiece determines its current task and all forward successors of the machine it has just left. The workpiece then sends an invitation to bid to all these machines. This call always includes the current workpiece task.

*Step 2.* If a machine  $M$  receives an invitation to bid for a current task  $\vec{o} = (o_1, \dots, o_n)$ , it checks whether or not it is able to perform a subtask of  $\vec{o}$ . This can be done by a simple check on  $o_1$ 's membership in  $op(M)$ . If this is the

case, then  $M$  issues a bid; otherwise, it simply ignores the call. Short-term disturbances of some of  $M$ 's operations are ignored here. This is because the subject of the negotiation is a *future* allocation of a subtask of  $\vec{o}$  and the current situation obviously does not tell us much about a machine's state when the workpiece enters the machine.

$M$  issues no bid without making sure that it is actually ready to accept a new workpiece; it therefore checks whether  $P^M < P_{max}^M$ . If this is not the case, then it does not answer the call either. A bid of a machine  $M$  always includes (a) the current size of  $M$ 's virtual buffer and (b) the maximal subtask of  $\vec{o}$  which contains only operations of  $op(M)$ .

*Step 3.* The workpiece collects all of the bids for a specific call. If there is no bid at all, then the workpiece issues another invitation to bid, continuing with step 1; otherwise the workpiece awards one of the bids. The workpiece compares the bids and awards the best bid. For this, both components (a) and (b) of a bid are relevant, with (a) having a higher priority. The lower the current size of the virtual buffer, the better. The more operations the maximal subtask (b) contains, the better. The workpiece awards the bid which is the best in this sense. The awarded machine then includes the relevant workpiece in its input buffer.

Usually, a number of auctions of this kind take place simultaneously, even in an interleaved manner. Thus a single machine may participate in more than one auction at a time.

### 3.3. Violations of Principal Flow

As it stands, the auction protocol described thus far handles only the principal material flow. Let  $M$  be the machine that the workpiece just left. The auction protocol from above states that *forward* successors of  $M$  are invited to bid, while backward successors are ignored. Nothing seems to prevent us from inviting *all* admissible successors of  $M$  to bid, whether being forward or backward successors. When awarding a machine, forward successors can always be preferred in favor of backward successors. A backward successor may be awarded just in case there is no bid from a forward successor.

Such a strategy, however, may run into a deadlock: a number of machines may find themselves in a cyclic dependency, each waiting for another machine to be ready to accept a new workpiece. Consider two machines  $M_1$  and  $M_2$  being admissible successors of each other. Let  $w_1$  and  $w_2$  be workpieces residing in the output buffer of  $M_1$  and  $M_2$  respectively. Moreover, suppose that  $w_1$  may allocate its current task (or a subtask) to only  $M_2$ , while  $w_2$  may allocate its current task (or a subtask) to only  $M_1$ . If  $M_1$  and  $M_2$  are not ready to accept a new workpiece, then  $M_1$  and  $M_2$  actually find themselves in a deadlock.

In *West* such deadlocks are resolved as follows: A machine always bids if it has a subtask to offer, even if the machine is currently not ready to accept a new workpiece. Such a bid, however, is not made without including in it an explicit warning that the machine is currently not able to accept any new workpiece into its input buffer. When a workpiece awards a machine, it ignores any bid with such a warning. This rule is only violated if the following two conditions are met at the same time:

- (a) There is no bid from a forward successor.
- (b) There is no bid which does *not* contain a warning.

If these two conditions are met, a machine with a warning may be selected in step 3 of the auction protocol. Although the chosen machine is not ready to accept a new workpiece, it is awarded anyway. Such an award is always combined with a request to include the workpiece into the input buffer, irrespective of the current size of the virtual buffer. This is what we call an *enforcement award*.

Such an enforcement award may exceed the upper bound of a machine's virtual buffer. However, under certain, general conditions, this violation of an individual upper bound never results in a violation of the sum of the upper bounds at system level. Section 4 provides the details, including a formal proof.

### 3.4. Dynamic Routing

Once a workpiece has awarded a specific machine for further processing, the workpiece must be moved to its new goal. In a layout like the one depicted in Fig. 1, there is usually a vast number of different paths ultimately leading to the same goal. Of course, the shortest of these paths should always be preferred. More important than optimizing the routing, however, is the avoidance of any congestion, which may have disastrous consequences on the performance of the overall system. In an unpredictable environment like a manufacturing system, jams can be avoided only by strictly separating the actual routing from the goal itself.

In *West*, such dynamic routing is ensured through a sequence of bilateral communications, each time between the workpiece and the next switch it approaches. A switch always tries to move a workpiece directly to its goal, thus trying to optimize the routing. If an exit is not available, then an alternative route is taken. In this case, however, the priority of the workpiece is incremented by exactly 1. These priorities are used to decide which workpiece to prefer if a switch has more than one possibility: the workpiece with the highest priority is always served first. This is to avoid indefinitely routing a workpiece along a cycle rather than to its actual goal.

## 4. Analysis

So far we have defined the behavior of the individual agents used in *West* and described in detail how these agents interact with each other. In this section we shall analyze the behavior which results at system level. Some observations are quite obvious; others give some deeper insights.

Let us begin with some simple observations. Whenever a machine breaks down, the *West* auction mechanism automatically diverts the material flow to other machines, thus balancing the machines' work in process. The actual material flow thus adjusts itself to the current situation. *West* has this important feature in common with other agent-based control mechanisms in the literature [8, 4, 3, 2].

In mass production, it is also important to enforce a material flow in the main manufacturing direction, while not sacrificing the possibility of violating this flow whenever it turns out to be inevitable. This is exactly what *West* does. When awarding a machine, forward successors are always preferred in favor of a backward successor; only if there is no bid from a forward successor, a backward successor may be awarded.

### 4.1. Self-Limiting System Behavior

But there is even more to *West*. What really is a unique feature of *West* is that it automatically adjusts itself to the current capacity bottleneck while avoiding congestions in the transport system. The system's current capacity bottleneck is automatically propagated in the opposite direction of the actual material flow; a process which continues until it has reached the system loader. As a result, the loader feeds only as many workpieces into the system as the system is currently able to handle.

To see how this works, let us again consider Fig. 4. A machine's virtual buffer can be thought of as a funnel. There is a stream of workpieces filling the funnel. This is the work in process or, as we called it, the input buffer. After being processed by the machine, a workpiece remains in the funnel until the workpiece has been included into the input buffer of another machine. This is the only way for a workpiece to leave the funnel. As soon as the number of workpieces in the funnel reach a certain level,  $P_{max}^M$ , the machine ceases to be ready to accept a new workpiece. We can think of this as a valve closing the stream filling the funnel. The input stream is thus automatically restricted to the maximal throughput of the output stream.

One might object that as soon as the buffer capacity of the transport system is at its limit the input stream is adjusted to the output stream anyway. However, a transport system which is at its physical limit is jammed. Jams may lead to a cascading loss of performance, persisting even long after the cause of the congestion has been resolved.

The point is therefore to achieve this linkage *without* forcing the transport system to its limit.

But this is not even the end to the story. There is a whole network of funnels and valves of the type described. The current capacity bottleneck propagates through this network, eventually reaching the system loader. An interesting feature of this kind of mechanism is that the topology of the network is not pre-defined; it is built up dynamically by the workpieces themselves. Take an arbitrary machine  $M$ . Every workpiece in  $M$ 's output buffer determines which machine would, in principle, be able to process the workpiece in its current state (see Step 2 of the auction protocol in section 3.2). This means that for the specific workpiece  $w$  there is an admissible material flow from  $M$  to all these machines. The network induced in this way is a subset of the admissible flow  $\prec$  depending on the specific product types currently in the system. The capacity bottleneck is always propagated in the opposite direction of this subset of the admissible flow, no matter how it looks like. The *West* mechanism can thus be regarded as being *self-organizing*.

## 4.2. An Upper Bound on the System Load

In what follows we shall make this self-limiting system behavior more precise. If the admissible flow of material is acyclic, then the *West* mechanism will obviously never violate any of the upper bounds  $P^{M_i} \leq P_{max}^{M_i}$ . This is no longer true if the admissible flow is cyclic. In this case, an enforcement award may indeed violate a machine's individual upper bound. A layout such as the one depicted in Fig. 1 is able to cope with such a local violation, at least as long as the total capacity of the transport system is not exceeded. This is exactly what the *West* mechanism guarantees: a violation of a machine's individual upper bound is always canceled out at system level. In particular, for a specific subset  $\mathcal{M}$  of machines, the overall upper bound  $\sum_{M \in \mathcal{M}} P^M \leq \sum_{M \in \mathcal{M}} P_{max}^M$  will never be violated, even if some of the individual bounds *are* violated. The subset of machines relevant here are all those machines which are involved in a what we call a feedback loop.

**Definition 6** *Let  $M_0$  be a machine. A feedback loop starting at  $M_0$  is a sequence  $M_0, \dots, M_n, M_0$  of machines ( $n \geq 0$ ) such that for every  $i$  ( $0 \leq i < n$ ),  $M_{i+1}$  is a forward successor of  $M_i$ , while  $M_0$  is a backward successor of  $M_n$ .*

Take the admissible flow of material depicted in Fig. 3. The reader may check that it contains exactly four feedback loops.

In order for the *West* mechanism to be able to maintain an upper bound at system level, feedback loops have to obey a special structure.

**Definition 7** *A feedback loop  $M_0, \dots, M_n, M_0$  has a single point of entry iff for every machine  $M \notin \{M_0, \dots, M_n\}$  and*

*every  $M_i \in \{M_0, \dots, M_n\}$  with  $M \prec M_i$ , it holds that  $M_i = M_0$ .*

Roughly speaking, this means that  $M_0$  is the only port through which the admissible flow may enter the feedback loop. The four feedback loops of Fig. 3 all have a single point of entry.

**Theorem 1** *Let  $M_0$  be an arbitrary machine. Let  $\mathcal{M}$  be the set which, in addition to  $M_0$ , contains all machines involved in at least one feedback loop starting at  $M_0$ . Suppose that all these feedback loops (if there is any) have a single point of entry. Then the *West* mechanism maintains the following upper bound:*

$$\sum_{M \in \mathcal{M}} P^M \leq \sum_{M \in \mathcal{M}} P_{max}^M. \quad (2)$$

**Proof.** Let us begin with the case where there is no feedback loop starting at  $M_0$ , so that  $\mathcal{M} = \{M_0\}$ . According to the definition of a feedback loop, this means that there is no machine  $M'$  which has  $M_0$  among its backward successors. Consequently no workpiece will ever select  $M_0$  for an enforcement award. Any bid from  $M_0$  is thus ignored, unless  $P^M < P_{max}^M$ . This guarantees that the upper bound  $P^{M_0} \leq P_{max}^{M_0}$  will never be violated, as was to be shown.

Let us continue with the case where there is at least one feedback loop starting at  $M_0$ . Assume that  $M_0, \dots, M_i, M_0$  ( $i \geq 0$ ) is one of these feedback loops. According to the definition of a feedback loop,  $M_0$  is a backward successor of  $M_i$ . A workpiece  $w_i$  residing in the output buffer of  $M_i$  can therefore select  $M_0$  for an enforcement award, possibly violating  $M_0$ 's individual upper bound  $P^{M_0} \leq P_{max}^{M_0}$ . Such an enforcement award moves  $w_i$  from the output buffer of  $M_i$  to  $M_0$ 's input buffer. For  $M_i = M_0$  this enforcement award would obviously not alter the size of  $P^{M_0}$ , thus not violating  $M_0$ 's individual upper bound either. So let us consider the case  $M_i \neq M_0$ . In this case,  $w_i$ 's enforcement award leaves  $P^{M_0} + P^{M_i}$  unchanged. As both  $M_0$  and  $M_i$  are contained in  $\mathcal{M}$ ,  $\sum_{M \in \mathcal{M}} P^M$  is not changed either.  $P^{M_0} + P^{M_i}$  might only be increased if, after  $w_i$ 's enforcement award, another workpiece awards  $M_i$ . Let us assume that  $w_j$  awards  $M_i$ . Let us, moreover, assume that at the time  $w_j$  awards  $M_i$ ,  $M_j$  is the machine which hosts  $w_j$  in its output buffer. This second award thus moves  $w_j$  from  $M_j$ 's output buffer to the input buffer of  $M_i$ . Such an award might increase  $P^{M_0} + P^{M_i}$ .

If  $M_j \in \mathcal{M}$ , then  $w_j$ 's award clearly does not change  $\sum_{M \in \mathcal{M}} P^M$ . Consequently  $w_j$ 's award cannot increase  $\sum_{M \in \mathcal{M}} P^M$ , unless  $M_j \notin \mathcal{M}$ . However, if  $M_j$  was not a member of  $\mathcal{M}$ , then  $M_0, \dots, M_i, M_0$  would not have a single point of entry, contradicting the assumption that every feedback loop starting at  $M_0$  has a single point of entry. In fact,  $w_j$  cannot award  $M_i$ , unless  $M_i$  is an admissible

successor of  $M_j$ . Now,  $\mathcal{M}$  contains all machines occurring in at least one feedback loop starting at  $M_0$ , thus it also contains all machines occurring in the specific feedback loop  $M_0, \dots, M_i, M_0$ . This means that  $M_i \in \mathcal{M}$ . But then,  $M_j \notin \mathcal{M}$  would violate the assumption that  $M_0, \dots, M_i, M_0$  has a single point of entry; there would be at least two points of entry, namely  $M_0$  and  $M_i$ .

This shows that  $w_i$ 's enforcement award leaves  $\sum_{M \in \mathcal{M}} P^M$  unchanged, thus never violating (2).  $\square$

**Corollary 1** *Let  $\mathcal{M}$  be the set of all machines of a manufacturing system. Suppose every feedback loop (if there is any) has a single point of entry. Then the West mechanism maintains the upper bound (2).*

### 4.3. Freedom of Deadlocks

Distributed mechanisms like *West* are always in danger of running into a deadlock. In manufacturing control, such an incident may cause the complete standstill of a line, thus leading to a high economic loss. The *West* mechanism is guaranteed to be free of deadlocks. In what follows we shall give a formal proof of this fact.

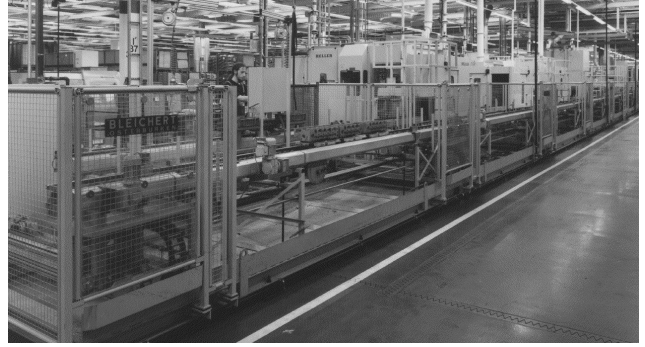
**Definition 8** *We say that a machine  $M_1$  blocks another machine  $M_2$  iff there is at least one workpiece in  $M_2$ 's output buffer which, if  $M_1$  was ready to accept a new workpiece, could in principle award  $M_1$ . This relation induces a directed graph of all current blockings. There is a deadlock between two machine agents iff this graph contains at least one cycle in which both machine agents occur.*

**Theorem 2** *The West mechanism is guaranteed to be free of deadlocks.*

**Proof.** The proof proceeds by *reductio ad absurdum*. Assume that there is a deadlock between two machines. This assumption will be shown to lead to a contradiction. A deadlock between two machines means that both machines occur in at least one sequence of machines  $M_0, \dots, M_n, M_{n+1}$ ,  $n \geq 0$ , such that (a)  $M_{n+1} = M_0$  and (b) for every  $i$  ( $0 \leq i \leq n$ ),  $M_i$  blocks  $M_{i+1}$ . First observe that  $M_i$  may block  $M_{i+1}$  only if  $M_i$  is a forward successor of  $M_{i+1}$ ; if  $M_i$  was a backward successor of  $M_{i+1}$ , then a workpiece in  $M_{i+1}$ 's output buffer could only award  $M_i$  by an enforcement award. Therefore, (b) implies that  $M_{n+1} \prec_p M_n \prec_p \dots \prec_p M_0$ . As  $M_{n+1} = M_0$ , this would mean that the principal flow is cyclic. According to Definition 2, however, a principle flow is always acyclic.  $\square$

## 5. Evaluation

In order to evaluate the new approach, we have conducted a series of simulations, all of which are based on



**Figure 5. DaimlerChrysler's Prototype.**  
(Source: BLEICHERT Osterburken)

authentic product types and cycle times. The disturbance characteristics have been taken from existing machines. A typical configuration consists of 4 blocks of identical machines. The number of machines in a block ranges from 5 to 11, 36 machines in total. The simulations have shown that the *West* mechanism is extremely robust against disturbances of machines as well as failures of control units. Its performance is nearly optimal. The following table summarizes the outcome of a characteristic simulation run.

Running time	Number of machines	Average throughput	Theoretical optimum
63 h	36	70.54 workpieces/h	70.73 workpieces/h

The *West* mechanism thus achieves about 99.7% of the theoretical optimum.

We also tested a hypothetical production process with different product types of lot size 1, where the operations are distributed over the machines in an irregular fashion. Even under such unusual conditions performance is still satisfactory. However, the complexity of the production process prevents us from computing the exact theoretical optimum here and, thus, also from conclusively assessing *West*'s performance in this particular setting.

DaimlerChrysler installed the new control approach as a bypass to an existing large-series manufacturing line for cylinder heads. The bypass, located in a plant in Stuttgart-Untertürkheim, Germany, is shown in Fig. 5. The layout and the control system are basically as described in this paper. Based on a DaimlerChrysler specification, Schneider Electric implemented the agent system and put it into operation [7].

The bypass has undergone a series of performance tests. These tests showed that the results of the simulations are still valid under real manufacturing conditions and, thus, demonstrated the benefits and industrial feasibility of the approach right from the start.

## 6. Conclusion

In the sense of [5], manufacturing control is a *going concern*. The goal here is to continuously optimize the throughput rather than to solve a static problem where all relevant parameters are given from the start. Manufacturing control is thus a kind of iterated game against nature [10], where the control system must cope with highly unpredictable events like resource disturbances and product changes. Consequently, our approach is to dynamically assign resources rather than computing optimal schedules, which would be invalidated by constantly changing parameters anyway. We argued in favor of an auction-based approach, where workpieces auction off their current task while machines bid for a subtask. When awarding a machine a workpiece takes into account not only the machine's current work in process, but also the outgoing flow of material; that is, all those workpieces which already have been processed by the machine without yet being able to award an appropriate new machine. If a machine's outgoing stream is blocked, then eventually the machine will not accept any new workpiece, thus blocking its input stream as well. In this way, a capacity bottleneck is automatically propagated in the opposite direction of the actual material flow. This kind of self-limitation is established in precise mathematical terms and so is the deadlock-freedom of the approach.

From a chaos-theoretical point of view, a manufacturing system is a complex non-linear system. It is non-linear in that a break-down of a machine may cause a jam in the transport system, persisting long after its cause has been resolved. In a flexible manufacturing system with multiple machines offering the same operation, the possibility of a disruption decreases dramatically. However, the fundamental fact that a disruption may cause further disruptions still remains in effect. This type of negative feedback may force the system into an unfavorable chaotic regime. An optimal control system should avoid these negative feedbacks, while exploiting the flexibility provided by redundant machine capabilities. This is exactly what our auction-based approach does. Self-limitation guarantees that the system never exhibits a chaotic regime, while late binding of resources exploits the flexibility within the remaining state space.

An industrial prototype proved the feasibility and the benefits of the new approach. The prototype basically implements the mechanism described in this paper. One essential ingredient, however, is not presented here. As defined in this paper, the auction mechanism does not guarantee fairness. By *fairness* we refer to the fact that after a finite number of rounds, every workpiece is able to award a machine. There are two different ways to guarantee fairness. The first option is to provide each workpiece with a certain budget. The workpiece with the highest budget will have the privilege of awarding a machine. When a workpiece awards a

machine, it has to pay a certain tax to all other workpieces claiming the same resource. This procedure resembles the Clark tax, see e.g. [6, p. 208ff]. Another way to guarantee fairness is by managing special waiting queues for awards, one for each machine. The latter approach enables us to implement the system in a strictly decentralized manner, while the first pre-supposes a central auction board on which all demands and bids are displayed. This will be discussed elsewhere.

**Acknowledgements.** We would like to thank our colleagues Hartwig Baumgärtel, Sven Brückner, and Harald Windisch for their invaluable contributions to the design and evaluation of the system. We would also like to thank our colleagues Christian Anders, Christoph Siegel, and Gerold Winz from the plant in Stuttgart-Untertürkheim for launching the project. Without their full commitment, this project would never have been crowned with success.

## References

- [1] S. Bussmann and D. McFarlane. Rationales for holonic manufacturing control. In H. Van Brussel and P. Valckenaers, editors, *Proceedings of the 2nd Int. Workshop on Intelligent Manufacturing Systems*, pages 177–184, 1999.
- [2] T. Kis, J. Vancza, and A. Markus. Controlling distributed manufacturing systems by a market mechanism. In *Proceedings of the 12th European Conference of AI (ECAI'96)*, pages 534–538, 1996.
- [3] J. Maley. Managing the flow of intelligent parts. *Robotics and Computer-Integrated Manufacturing*, 4(3/4):525–530, 1988.
- [4] H. Parunak, J. White, P. Lozo, R. Judd, B. Irish, and J. Kindrick. An architecture for heuristic factory control. In *Proceedings of the American Control Conference*, pages 548–558, 1986.
- [5] H. V. D. Parunak. From chaos to commerce: Practical issues and research opportunities in the nonlinear dynamics of decentralized manufacturing systems. In *Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems*, pages k15–k25, 1999.
- [6] T. W. Sandholm. Distributed rational decision making. In G. Weiss, editor, *Multiagent Systems*, pages 201–258. The MIT Press, 1999.
- [7] R. Schoop and R. Neubert. Agent-oriented material flow control system based on DCOM. In *Proceedings of the 3rd IEEE International Symposium on object-oriented real-time distributed computing (ISORC 2000)*, 2000.
- [8] M. Shaw and A. Whinston. Task bidding and distributed planning in flexible manufacturing. In *Proceedings of the 2nd Conference on AI Applications*, pages 184–189, 1985.
- [9] H. R. Varian. Economic mechanism design for computerized agents. In *Proceedings of the First USENIX Workshop on Electronic Commerce*, pages 13–21, 1995.
- [10] P. Wegner. Interactive foundations of computing. *Journal of Theoretical Computer Science*, 192:315–351, 1998.