Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

A FLEXIBLE ASSEMBLY GLOBAL CONTROL SIMULATION

Thomas Ernst

FAW Ulm
Research Institute for Applied Knowledge Processing
Helmholtzstraße 16
89081 Ulm/Donau, GERMANY

Avetik P. Matevosian

IISA
Interministry Institute of Skill Advance
Acharian Street 31
375040 Yerevan, ARMENIA

ABSTRACT

In this paper we present a concept for the simulation of a globally controlled flexible assembly manufacturing system. The objective was to study the runtime behaviour of the global control system and to integrate the medium term planning and local shop floor scheduling in the simulation model. A scheme for the supplier-manufacturing interaction based on a pull strategy is introduced. The model was developed using BLOCS/M, a discrete-event and object-oriented simulation system. The implementation of the model is shown by means of an illustrative example of an electromotor factory.

1 INTRODUCTION

With the tendency of the industrial development towards the production (manufacturing and buying) in very small quantities, the application of flexible assembly/manufacturing systems becomes more and more important. The flexible manufacturing systems, as a rule, are formed from cells of machines, ensuring the fabrication of the distinct "family of parts" from a variety of different input parts and raw materials. Each cell also includes associated stock points connected by the transportation facilities of the material handling system.

Many papers, for example, (Forestier, 1981; Pratt, 1982; Sukhodolski, 1986; Zakharian, 1989) are devoted to the analysis and optimization of the manufacturing system layout and buffer sizes, simulation of pallet transportation and equipment operation. Numerous investigations (Akella, 1987; Buzacott, 1992; Glassey, 1990) have directed their attention towards the solution of shop floor control problems, including development and simulation of different dispatching rules and scheduling algorithms.

The objective of this study was to develop a concept for a flexible assembly simulation which integrates the overall control of the whole technological equipment and the coordination between suppliers and manufacturing system. This coordination problems are very important at the conditions of a big range of finished products, based on small lots of a high variety of different components. In order to respond to incoming orders for completed products and at the same time to reduce the extra inventories at the input stores, the control system should be able to efficiently predict the current demand for particular input parts and materials and to coordinate the manufacturing process according to the current orders. To model the real control system it was also necessary to implement a pull strategy, according to which materials and parts from suppliers are received, instead of having only the default push strategy.

The developed concept and the simulation model have been validated in a case study of an electromotor factory. It is shown how the inventory and the production vary under changing control parameters.

2 THE COORDINATION SYSTEM

The purpose of the coordination system considered in this paper is to provide instructions to each of the workstations concerning the part types and quantities to be produced over a distinct time period. The second function is to determine the amount and release dates of input parts and materials, based on the incoming orders for finished products. This includes sending requests to the corresponding outside suppliers.

In order to concentrate the attention on the above mentioned operational mechanism, we assume in our model, (see figure 1) that the processing sequence of the incoming orders has already been determined.

Before starting the description of the control algorithm in detail, we introduce a number of system parameters that determine the performance of the coordination system.

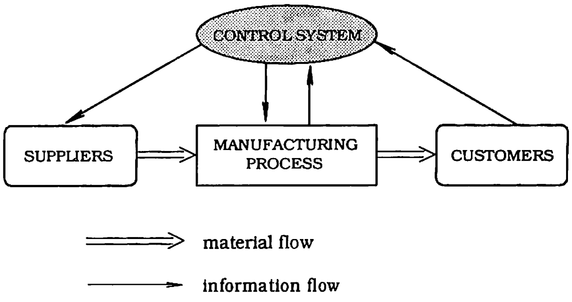

Figure 1. Block diagram of the model.

2.1 DeliveryTime

This is the delay between sending an order for input parts to the supplier and the time when this parts are supposed to be required. The *DeliveryTime* is specified for each supplier according to the actual delivery time. In particular it can be defined as equal to the mean delivery time. However, because of the uncertainty in actual flow times the materials often arrive earlier or later than necessary. A high value of *DeliveryTime* will result in undesirably large work-in-process inventories and the low one will lead to frequent downtimes of the technological equipment. Therefore the optimal values must be determined looking from the overall system point of view taking into consideration inventory costs against penalties associated with due date failurs.

2.2 ForecastingPeriod

This is the maximal period, starting from the current time, in which the coordination system forecasts a future demand in input parts and materials. The coordinator scans only those customer orders, which are anticipated to start in the forecasting time horizon. A request is sent to suppliers of the corresponding input products, for which the time offset is less than *DeliveryTime*. In particular the *ForecastingPeriod* must be greater or equal than the maximal *DeliveryTime*.

2.3 RevisionTime

Another parameter of the system, which must be determined during simulation is the review frequency of the coordinator. At the review points the coordinator sends requests to suppliers depending on the current state of the production process. The parameter *RevisionTime* which determines this frequency must be at most less than the minimal *DeliveryTime*.

The systems coordinator needs two additional parameters during the calculation of possible start times for the customer orders. The first one is called *Rate*, which is the time delay between successive outputs of two different units of a particular finished product at the end of the main assembly line.

The second parameter is called *CorrectionTime* and takes into account uncertainties of the entire facility as well as setup times required to start a new order for a different finished product.

The possible start time for a particular order is therefore calculated as follows, where i is the serial number of the order and *Pieces* is the quantity of finished products within the corresponding order:

$$StartTime_i = StartTime_{i-1} + Rate * Pieces_{i-1} + CorrectionTime$$

2.3 Algorithm

Figure 2 shows an overall control algorithm which is used to coordinate the starting and finishing of the orders, the initialization of workstations and operations and the definition and sending of orders to the outside suppliers.

The left and the middle branches of the algorithm (see figure 2) define the sequence of actions performed by the coordinator at each review point to generate forecasts of future demands in input parts and to release associated orders for outside suppliers. At the beginning of each review the coordinator reads the next waiting customer order (block 2) and calculates (block 3), according to (1), its possible start time. If the start time (block 4) is less than *ForecastingPeriod* coordinator defines the input parts and materials required for the completion of the given order, and if necessary generates a demand (blocks 5 - 10). This procedure is repeated until the start time of the next order in turn is greater than *ForecastingPeriod*. In this case coordinator checks whether there are parts to be ordered (block 11) and transmits an order to outside suppliers if needed (block 12). There can be a constraint that may restrict the minimal lot size delivered by the supplier. Therefore it can happen that the arriving lot contains a certain quantity of superfluous parts or material. Coordinator takes into account this information (blocks 9 and 10) when generating a future demand.

The coordination system ensures also the supervision of the technological equipment (right branch of the algorithm). This includes the definition of the current orders for workstations concerning the product types and their quantities to be produced. No order can be started on a workstation without receiving a special instruction. Coordinator (see block 13) awaits an acknowledgement of the completion of the current order in turn. Based on the available customer orders the coordinator defines new tasks for workstations, including the information concerning the part types and required quantities. To make the appropriate decisions, coordinator continuously maintains the information concerning the waiting and already started orders, which were read from the input file (see block 2).

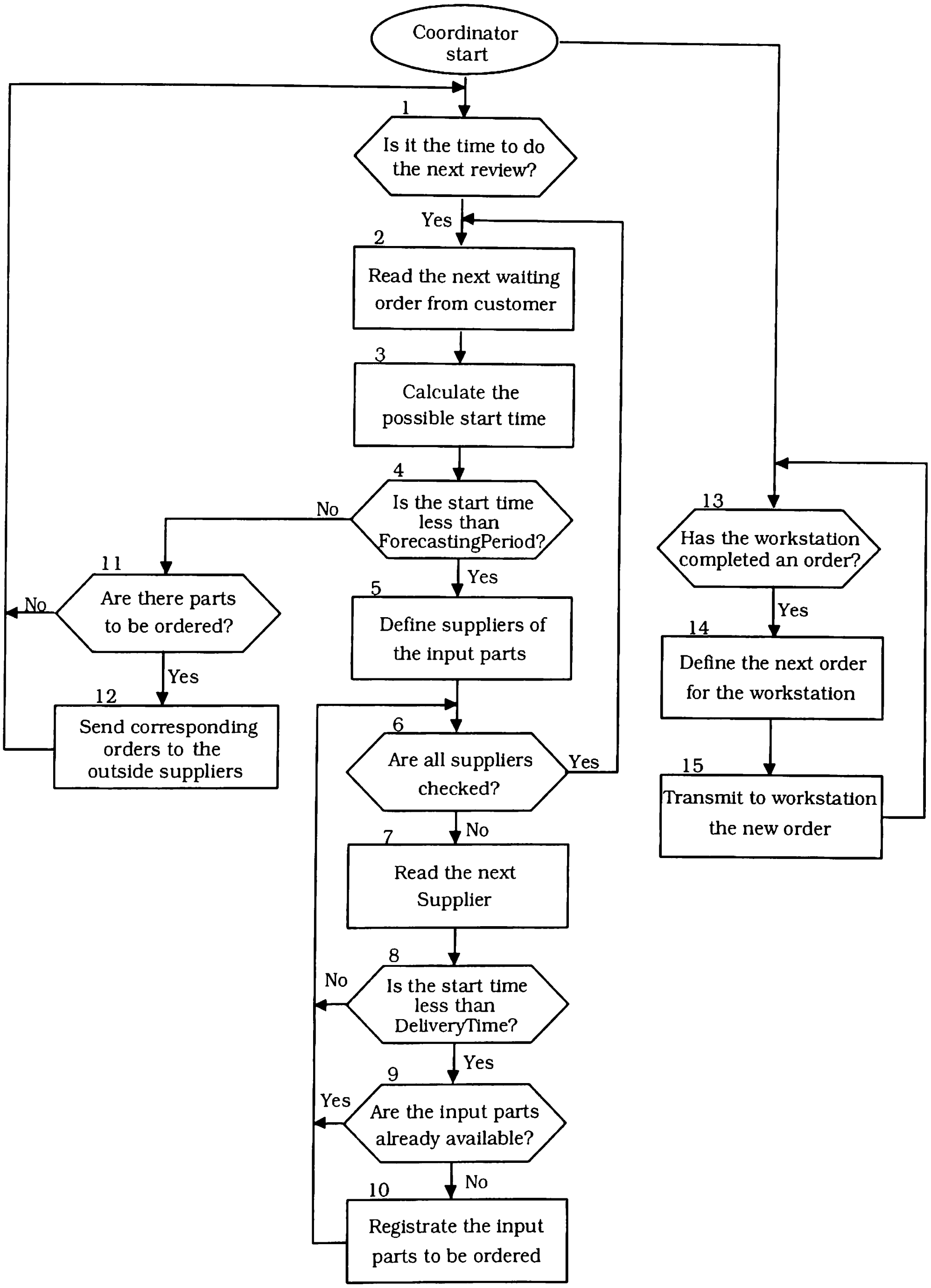

Figure 2. An overall control algorithm.

3 THE MODEL

The simulation model described in this paper is developed using BLOCS/M, the "Berkeley Library of Objects for Control and Simulation of Manufacturing" (Glassey, 1989), an Objective C based simulation system.

In terms of object-oriented programming (Cox, 1987) each module of the simulation system is implemented as a class. Each class defines a number of variables and procedures, called methods, which determine the behaviour of objects belonging to the class. The interaction between different objects is being realized by the exchange of messages. The objects in BLOCS/M represent workstations, queues, operations, processing sequences, lots, lot generators etc. An object called *Coordinator* monitors all important events in the model. The discrete event timing is realized by the *Timer* object. To start the simulation a set of userdefined data is specified in the input data files.

To implement the coordination mechanism described above and to enable the simulation of the assembly processes, a number of extentions to the BLOCS/M system were made. The following is a short description of this extentions.

3.1 Splitting and Joining of Lots

The sequence of processing steps a lot of a particular product has to follow through the factory is specified by means of *Route* or *OperationSequence* objects. Each processing step is performed at a unique workstation. In addition to the data, used to define each of the operation sequences, a new data item concerning the splitting and joining of the lots at each step is added. The actual splitting and assembling takes into consideration the number of pieces of a particular lot.

3.2 Assembling

To enable the simulation of assembly processes three new classes were added, which are *AssemblyOperation, AssemQueue* and *AssemblyWorkstation*. The *AssemblyOperation* is a static data object defining an assembly operation. Apart from the ordinary information about operation duration, setup etc, each *AssemblyOperation* object defines coproducts and a new assembled product. The *AssemblyWorkstation* represents an object which is composed of machines that perform the same set of assembly operations. When two parts are assembled together, the lots associated with this parts are freed and one new lot of the assembled product is created. The parts to be assembled are taken from the *AssemblyQueue* which consists of a number of subqueues, one for each coproduct. The current *AssemblyOperation* is determined dynamically during simulation. This can be realized in off-line mode, where the workstation makes the decision, or in on-line mode, where the global control system assigns the assembly operation as well as the quantity and type of the manufactured product to the workstation (see figure 2, blocks 14 and 15).

The most important feature of the *AssemblyWorkstation* is the creation of a new product from one or several coproducts. This concept is also useful for modeling a non-assembly workstation, which can produce different part types from the same input material.

3.3 Suppliers

To create lots entering the factory, two new objects of type *Supplier* were introduced. Both *Supplier* objects create and deliver new lots only after receiving a request from the coordinator, which represents a pull strategy in material flow. In one case, the decisions made by the coordinator are based on the current level of inventory in the system, in particular by simply checking the value of the safety stock in the input store. In the second case the coordinator forecasts the future demand in input parts and materials using the algorithm briefly discussed in the previous section.

The information concerning the customer orders is entered via an input data file, which coordinator reads dynamically during simulation. The information associated with each order includes the order name, an identifier of the finished product, the required quantity of pieces and also may include information concerning the due dates and other parameters. Each finished product has a list of suppliers for the corresponding input parts.

4 EXPERIMENT

In this section we demonstrate the application of the developed model to improve the productivity of the simulated assembly manufacturing system by appropriate coordination parameters.

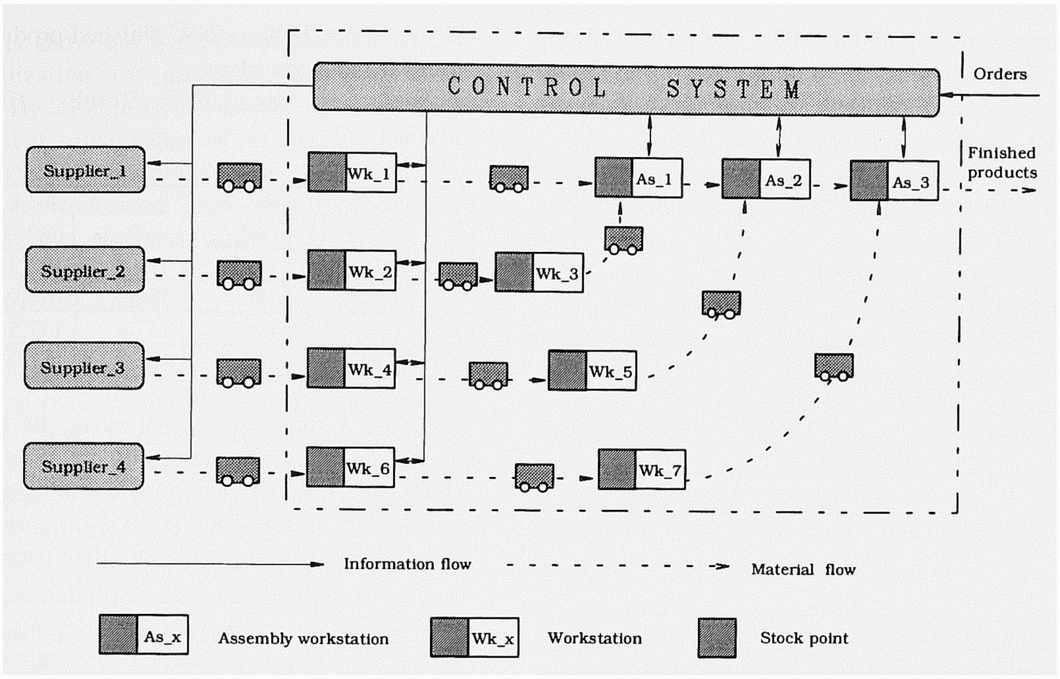

Figure 3. Manufacturing system

The simulated manufacturing system (see figure 3) represents one fragment of an electromotor factory. A fragment includes ten workstations ensuring respectively: Wk_1 - fabrication of windings; Wk_2 and Wk_3 - production of stator sheets and stator cores; Wk_4 and Wk_5 - pressure casting and the machining of the stator frames; Wk_6 and Wk_7 - manufacturing of the rotor shafts and the final processing of rotors; As_1 - dropping of the winding in the slots of the stator core; As_2 - assembling of stator frames with stators, and As_3 - installation of the rotor. Some of the workstations listed above represent aggregated models of production lines and cells of the real manufacturing system. The input materials are coming from four outside suppliers, delivering respectively the magnet wire, electrical sheet steel, alluminium and calibrated steel. Depending on the request, two different types of magnet wire can be delivered. According to the incoming orders from customers, four different finished products: Motor_1, Motor_2, Motor_3 and Motor_4 can be produced in this example. As can be seen from figure 3, the control information is sent only to workstations Wk_1, Wk_2, Wk_4, Wk_6, As_1, As_2 and As_3, because they either communicate with suppliers or perform an assembly operation. The input data specifying the assembling of each finished product is given in table 1.

Table 1. Definition of finished products (workstation sequence and coproducts)

Finished Product	Workstations	Coproduct
Motor_1	Wk_1, Wk_2, Wk_4, Wk_6, As_1, As_2, As_3	winding1, core1, frame1, rotor1, stator1, subassembly1, motor_1
Motor_2	Wk_1, Wk_2, Wk_4, Wk_6, As_1, As_2, As_3	winding1, core1, frame2, rotor1, stator1, subassembly2, motor_2
Motor_3	Wk_1, Wk_2, Wk_4, Wk_6, As_1, As_2, As_3	winding2, core2, frame1, rotor2, stator2, subassembly3, motor_3
Motor_4	Wk_1, Wk_2, Wk_4, Wk_6, As_1, As_2, As_3	winding2, core2, frame2, rotor2 stator2, subassembly4, motor_4

A number of simulation runs were made using different input parameters, associated with the suppliers, such as *DeliveryTime* for supplier_1 and safety stock values for others (see table 2).

Table 2. Changing Input Parameters in all Simulation Runs

Run	Supplier_1 (minutes)	Supplier_2 (units)	Supplier_3 (units)	Supplier_4 (units)
1	550	145	100	125
2	620	170	115	140
3	690	195	130	160
4	770	215	140	175

Table 3. Constant Input Parameters in all Simulation Runs

Mean Production Rate	8 minutes
Time to deliver magnet wire	660±110 minutes
batch size	order dependent
Time to deliver sheet steel	1440±280 minutes
batch size	360 units
Time to deliver alluminium	960±210 minutes
batch size	240 units
Time to deliver calibrated steel	1200±170 minutes
batch size	300 units

The other input parameters such as processing and setup times, reliability and repair times of workstations were also held constant in the model.

In addition, an input "orders.file" is specified. Table 4 shows a part of this file.

Table 4. Examples of Customer Orders

Order	Finished product	Quantity
order_1	motor_1	35
order_2	motor_4	5
order_1	motor_3	40
order_1	motor_1	15
order_1	motor_2	45

Table 5 gives the mean inventories in four input stores of the system respectively for each of the simulation runs after 352 hours of the system operation. As can be seen, the amount of inventory continually increased from simulation run 1 to 4 as a result of changing parameter values given in table 2. However, this is accompanied by the increase in production of the system from 2340 to 2492 finished products or from 54 to 63 completed orders.

Table 5. Simulation Results (Inventory and Production)

Run	Stock_1 (Wk_1)	Stock_2 (Wk_2)	Stock_3 (Wk_4)	Stock_4 (Wk_6)	Production
1	21.6	172.4	112.8	137.3	2340
2	27.5	180.4	124.5	143.9	2449
3	36.7	199.5	137.4	157.5	2490
4	37.9	220.1	146.5	168.9	2492

Figure 4 shows the variation of the inventory over the time in two input stores of the system for the first simulation run. As can be seen, the input stores are empty over certain time periods because the lots from suppliers are coming often later than necessary.

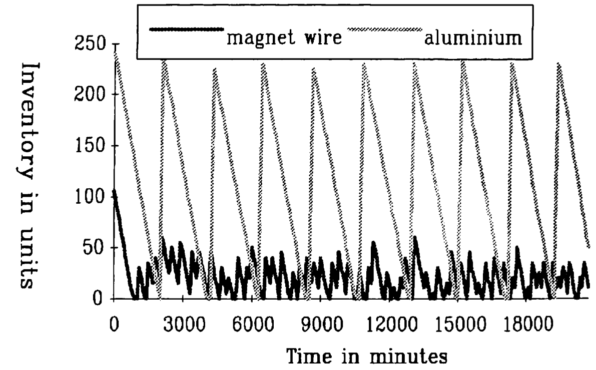

Figure 4. Level of inventory for run 1.

In contrast, the input parts in run 4 are always available as a result of having materials from outside suppliers earlier than necessary.

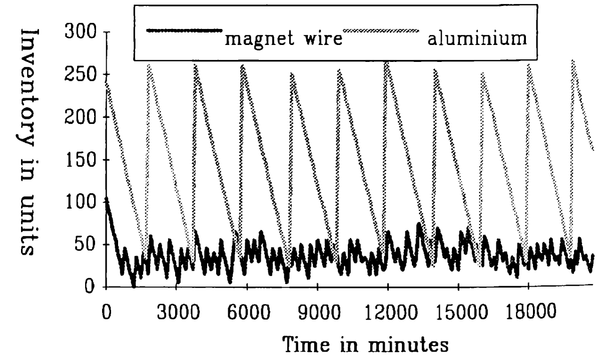

Figure 5. Level of inventory for run 4.

These results show the trade-off between inventory costs and utilization for different parameter value combinations. By an appropriate analysis of simulation results the most effective values of the above mentioned parameters can be determined by weighting the inventory and transportation costs against penalties of the missing due dates and being tardy.

5 FUTURE RESEARCH

In this paper we have introduced an approach for the simulation of an overall control of flexible assembly / manufacturing systems. The application of the developed model allows to improve the interaction between manufacturing system and outside suppliers through the definition of safety stock values, delivery times and batch sizes in which the input materials are delivered. The simulation model can be used to analyse and determine the processing sequences of orders from customers, taking into account their due dates and required setup times.

The efficiency of the model can be substantially increased by developing and implementing optimization algorithms ensuring the on-line sequencing of the incoming orders. In this case the orders are not specified in an input file, but are generated dynamically during simulation according to the available statistics. In the case of long delays between incoming orders, the system may generate forecasts of future demands in finished products. Of great importance is also the incorporation of a special work-in-process inventory control mechanism based on a pull logic.

REFERENCES

Akella R. 1987. *Real time part dispatching in flexible assembly test and manufacturing system.* Technical Report. Carnegie Mellon University, Pittsburg.

Cox B.J. 1987. *Object-Oriented Programming. An Evolutionary Approach.* Addison-Wesley.

Buzacott J.A., J.G. Shanthikumar. 1992. A general approach for coordinating production in multiple-cell manufacturing system. *Production and Operation Management* 1: 34-52.

Forestier I.P. 1981. A simulation program for the analysis and design of a flexible manufacturing system: *Proceeding of the 1-st International Conference on Cybernetics and Society.* 257-259. Atlanta.

Glassey C.R. 1989. *An overview of BLOCS/M: Berkeley Library of Objects for Control and Simulation of Manufacturing.* Department of Industrial Engineering and Operations Research, University of California, Berkeley 94720.

Glassey C.R. 1990. *Real time scheduling of batch operations.* FAW-TR-91007, Ulm.

Pratt C.A. 1982. A Simulation tool for manufacturing. *Simulation* 39: 143-144.

Sukchodolsri A.M., O.V. German, J.P. Bejtiuk. 1986. The use of simulation in the design of flexible manufacturing systems. *Devices and Control Systems* 4: 10-14. Moscow.

Zakharian S.S., A.P. Matevosian. 1989. A modular hierarchical simulation of manufacturing systems: *Preprints of INCOM'89, 6-th Symposium on Information and Control Probems. in Manufacturing Technology.* 559-562. Madrid.

AUTHOR BIOGRAPHIES

THOMAS ERNST is a research scientist at the FAW Ulm (Research Institute for Applied Knowledge Processing). His research interests lie in the area of combinatorial optimization and computer simulation including application studies as well as further development of methodologies and tools.

AVETIK P. MATEVOSIAN is a senior scientist at the Interministry Institute of Skill Advance in Yerevan, Armenia. His research interest is the control and simulation of manufacturing systems and communication networks. Since 1990 he has been visiting scientist at different german companies and institutes including the FAW Ulm.