A Market Approach to Holonic Manufacturing

A. Márkus, T. Kis Váncza, L. Monostori (2), Computer and Automation Research Institute,
Hungarian Academy of Sciences, Budapest, Hungary
Received on January 8, 1996

Abstract

The concept of holonic manufacturing is based on the cooperation of autonomous, functionally complete entities with diverse, often conflicting goals. The paper introduces a market mechanism for coordinating the activities of intelligent agents that pursue their own interest by operating under bounded rationality in a changing, hardly predictable environment. The market model is used for solving dynamic order processing and scheduling problems: conflicts between local scheduling agents are resolved by negotiating and bargaining on simple common terms of tasks, due dates and prices.

Keywords: Holonic Manufacturing, Dynamic Scheduling, Artificial Intelligence

1. Introduction

The evolution of manufacturing systems towards *cooperative manufacturing* complements the efforts made for the realization of *computer integrated manufacturing* [2, 6]. With all of its merits, integration resulted in rigid, hierarchical control architectures whose structural complexity grew rapidly with the size of the systems and the variety of production. Moreover, integration leads to complex decision problems [17]. Disturbances in manufacturing, as well as changing demand for certain products provides an unstable environment: it is next to impossible to be prepared with pre-programmed, top-down responses to abrupt changes, or to complete computations on sophisticated decision models before the results themselves are invalidated. Sudden changes can be responded only if decision rights are co-located with information; if time is considered as a limiting resource of decision-making, and if the system has a redundant and easily reconfigurable organizational structure. Decentralization, however, introduces new elements of uncertainty into the control of manufacturing systems; these can be resolved by *cooperation* only.

Holonic manufacturing is a novel approach to cooperative manufacturing. By definition, a holonic manufacturing system consists of entities (holons) which are *autonomous* and *cooperative* [5, 15]. The control of a holonic system is distributed: based on their own situation assessment and knowledge, individual holons autonomously decide how to attune their actions with those of the others. On the other hand, holons are expected to cooperate: this means that they coordinate their actions in order to meet their own goals. Cooperativeness is an assumption that deserves further attention: should it be an intrinsic property of holons or should it be ensued by rules that govern the overall system? What can be done when global optimization criteria - such as meeting due dates, or maximizing profit - cannot be decomposed and distributed; when global criteria do not correspond to the local concern and interest of individual holons?

The main concern of the paper is cooperation in holonic manufacturing systems. First we define a production planning and control problem that integrates *dynamic order processing* [9] and *distributed job shop scheduling* [3, 14]. In this setting, products are manufactured by selfish, autonomous agents whose primary goal is to maximize their own profit. Profitable production requires an appropriate schedule of the tasks: there will be shown how such a schedule can be generated via a market mechanism that transforms the scheduling problem into an economical problem. Thanks to specific rules that govern the market, conflicts can be resolved for the benefit of the whole system.

The idea of negotiated factory scheduling emerged long before. Early attempts implemented dispatching mechanisms [10, 12] but did not concern advance scheduling. Conversely, [1] realizes a predictive scheduler with no reactive capabilities. For negotiation, versions of the contract net protocol [13] are used. Market-oriented programming based on general equilibrium theory has been applied to resource allocation problems with no temporal aspect [16]. Dynamic reconfigurability, which asserts itself again in the holonic concept, appeared first in [12] that suggested iterated bidding for selecting cells that complete a job. This approach was developed further into random manufacturing [8], where temporal coalitions of machines competed for incoming orders. Recently, a scheduling method based on Lagrangian approximation has been mapped into a holonic architecture [5].

2. Problem setting

In our model of manufacturing control two types of agents are defined: the first one incorporates a model of managerial activities in the factory, the other one the production related activities. These types of agents are called *Management* and *Machine*, respectively. In the present setting a single Management and a small number of Machine agents are used. In addition, there is a third party, the *Outside World*. Its identity is vague, but it can not be omitted when dealing with an economic entity.

The overall objective of the factory is to earn, in a long range time scale, as much profit as possible. Literally, *profit = income - cost*. The profit of Management is the payment received from the outside world minus the sum of payments given to the machines for working on the orders. The profit of a machine is the difference between payment from the management and its technological cost. Since the whole manufacturing system is considered as a single property, the share of profit among the agents is not taken into account as a control variable (no mechanism feeds the profit back into production for investments, agents do not go bankrupt, etc.), so the agents' only measure of their acting properly is their profit. To avoid extremities of selfish behavior and meet the requirements of common-sense manufacturing rationality the agents have to obey rules that define a market mechanism.

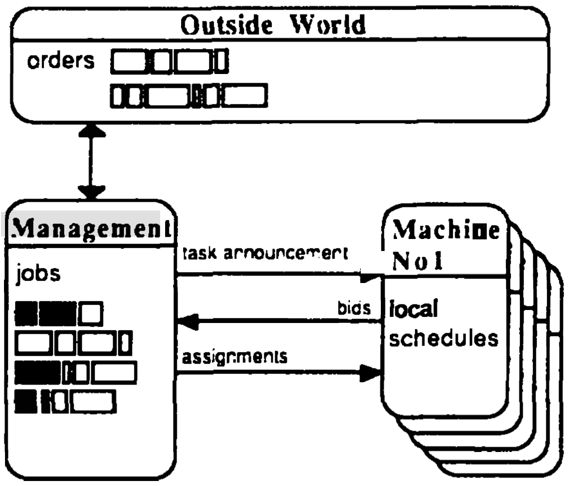

Fig.1. System configuration.

The outside world submits *orders* with given time and financial terms. Orders are sequences of *tasks*. Each task needs a specific *volume* of a technological *resource* at the machines. When the Management accepts an order, a *job* is created. To the tasks the Management attaches time and financial terms, and *announces* them to the machines. Machines prepare *bids* for the announcements. Bids accepted by the Management are called *assignments* (see Fig. 1).

If one assumes a predictable order stream handled by agents with unlimited computing power and ideal cooperation, the factory works in an optimal regime: the management accepts a mix of orders optimal with respect to the production capabilities and workload of the machines, and the machines work in a schedule that ensures that the factory as a whole earns maximal profit.

Our aim is to control the manufacturing system under more practical assumptions:

- No agent has an overall view on the state of the factory, (e.g., the management knows key parameters of the machines but it does not know how they make their detailed schedule). Interactions are restricted: the outside world contacts the management only, and the machines do not know the plans of each other.
- The agents have to decide in limited time (e.g., a good order has to be accepted as soon as possible, else another manufacturer may take it).
- The agents are self-interested as far as the regulations allow (e.g., to build up better working conditions for themselves, machines may deliberately bias the estimates on their future workload).
- The factory has to adapt itself to new situations in the changing world (e.g., a technological bottleneck may develop when well-paying orders show an increased need of a resource).

3. The activities of agents and the rules

The outside world and the order stream

The outside world submits a stream of orders; arrival of the orders is parallel with their processing. Orders may be rejected with no consequence, however, once accepted the factory must make them ready since the world commits itself to pay just for the finished orders. Each order has an ArrivalTime, a DueTime, a ContractPrice and a TardinessPenalty function. Price is fixed in terms of monetary units; the tardiness penalty is a function increasing in time. When the order is finished in due time, the outside world pays the ContractPrice to Management; finishing earlier has no financial effect. If the order is finished later than its DueTime then the ContractPrice is decreased by the appropriate penalty; the Management's income may be negative as well.

The future stream of orders is unknown, but its key parameters (such as the ratio of rush orders and their technological spectrum) are supposed not to change quickly.

Order evaluation and job formation

When an order arrives, Management decides whether to accept or reject it. The expected profit of the new order is estimated by linear regression on stored time and financial frames of similar tasks executed earlier.

If the Management accepts an order, a new job is created: it is described with its ArrivalTime, DueTime, ContractPrice and TardinessPenalty function. In addition, to each task Management computes its estimated time frame and manufacturing price; hereby each task is given as a five-tuple of the required Service, Volume, ArrivalTime, DueTime and Price; e.g., a task with (op1 20 40 66 40) means that 20 volume units of the service op1 are to be made in the absolute time frame of 40 to 66 minutes on the clock and the price payable to a machine is limited to 40 financial units. While Service and Volume are fixed, the other tags may be changed later.

The parameters of jobs are not released to the machines: jobs exist solely for the Management and the Machine agents' knowledge is limited to the level of tasks. This corresponds to the organizational principle of allocating to each agent comparable levels of knowledge, decision right and action.

Announcements of tasks

As soon as a job is formed, Management announces its first task to the Machines. The announcement contains the characteristic five-tuple of the task.

The next task of a job is announced when the preceding task starts on a machine. Due to assumptions below, finishing time of the preceding task is known at this time and it is equal to the arrival time of the next task.

The Machine agent and its bid preparation

The factory consists of a number of machines that have partially overlapping technological capabilities; some machines may be more suitable for certain or all operations. Each machine is given as a set of its technological capabilities, called resources here. Resources are described as triplets of Service, Speed and MinuteCost. E.g., executed on a machine with a resource of (op1 5 8), the task of making op1 for 20 volume units takes 4 minutes and its technological cost is 32 units.

The following machine related assumptions hold: (1) No task may be shared between machines. (2) No task may be delayed or abandoned after having started. (3) Costs of maintenance, tooling etc. are considered proportional to work volume and are included in technological cost. (4) Setup and further auxiliary costs are not taken into account. (5) Machines are supposed not to break down while making a task.

Some of the above limitations could be lifted with minor modifications of the model (3 and 4), but others, such as (1) are crucial. Most of the limitations above are rooted in our inability to attach financial equivalents to such factors. Though, in this market-based approach it makes not much sense to deal with, say, a breakdown event without considering its financial consequences and to regard it only as a modification of available resources.

Alternative advance schedules of the machines are represented in their *schedule trees*. The tree is known and updated exclusively by its owner. Before preparing a bid, a machine follows least commitment strategy: it tries to fit the announced task in all sensible ways into its schedule tree. Then bids are formed which contain StartTime, EndTime and Price of executing the task. Each machine may send more than one bid for an announcement or may leave the announcement unanswered. The bids must not hurt the financial or time constraints of the announcement.

Announcements are valid only for a given time (i.e., bids have to be submitted during that time) and at most one announcement can be in progress at any time.

Task assignments and their completion

Within a constant time after announcement, Management receives all the bids from the machines. If no bid has been received, it has to announce the same task again.

Management may accept several bids from several machines; usually they have *alternative, parallel assignments* for each task. After selecting a subset of the bids it notifies the Machines; accepted bids make new assignments at the machines.

The following rules apply to bids and assignments: (1) Management must not accept a dominated bid (bid B1 is dominated by B2 if B2 is better both with respect to its price and finishing time). This rule prevents the Management's hidden agreement with machines. (2) When a machine finishes its current operation, it has to select one from its startable assignments; it may go idle only if it has no assignment to work on.

When selecting an assignment, the machine may want to earn as much profit as possible now, or take into account its profit perspectives as well. The second objective is difficult to attain since a machine may deem a subtree of its schedule tree very promising, but later it may be unable to work along those branches since other machines might have started some of tasks there. There are problems with evaluating the free time of the machines, too (if orders are getting better, free time is getting more valuable). Accordingly, immediate profit is the chosen criterion of the selection of tasks to start.

When a machine starts to work on an assignment, it cancels assignments that turned infeasible and informs all other machines about the event so that they could prune their schedule trees accordingly.

Repeated announcements and lateness control of jobs

Management announces a task again whenever it has received no bid from the machines or when all assignments of the task have been canceled. The repeated announcement of the task may have different time and/or financial frame. Since announcements are made only when specific physical events happen at the factory floor, the announcement process may not become a loop.

Initially, the order acceptance mechanism allots suitable time frame to each task. However, due to delays at making previous tasks, Management may find that, under the current workload, the job would run out of time. In this case, Management marks the job as being a *late job* and starts to collect penalties from those machines that are technologically capable to produce the next task of the marked job. The total of penalties per time is equal to the penalty per time the Management will have to pay to the outside world. When a machine starts to work on the next task of a late job, the job ceases to be marked late, since the Management has already collected the penalty. If Management has collected more penalty for a job than it has to pay, it reimburses the machines.

This penalty mechanism belongs to the obligatory market rules. In this way Management becomes transparent: supposing that the order selection procedure is fair toward the machines, the overall efficiency of the factory depends only on the machines and the completion of the jobs becomes the machines' collective responsibility.

4. Social dilemmas among the machines

According to these market rules, machines can not form coalitions against Management and there is a fair competition for the well-paying tasks. It is risky to ask too much for a task, since another machine may offer a better bid. Normally this leads to moderate prices and drives the routine operation of the whole system.

Starvation of a job may occur when Machines are not willing to work on a task since they prefer more attractive tasks of other jobs. However, the penalties on late jobs impose a burden on all machines which are able to work on the next task. and this urges the machines to complete it as soon as possible. Anyway, individual interests may cross the system-wide objective of total profit maximization: in spite of the penalty, a rational selfish agent may decide to start another task which seems to be, all things considered, more profitable. Thus the individual can earn more, but the factory may suffer a loss. This is how social dilemmas [4, 7] appear in our model.

To resolve these type of conflicts, machines sometimes have to turn to less profitable tasks. Those that sometimes select less profitable tasks are called *cooperative machines*, while those that do not make this sacrifice are the *free riders*. Cooperativeness of a machine can be tuned to various degrees: it is expressed as the odds of selecting less profitable tasks. While there are no late jobs, machines can work selfishly and there is no need of cooperation; whenever a job seems to be late, cooperative behavior serves the common interest.

5. Implementation and experiments

We are experimenting with this market model in a simulated production environment that is implemented in the Common Lisp Object System (see Fig. 2). In order to account for the limited computing power of the agents, each agent records its computing time. The program works with a monitor that gives time slots to the agents in turn.

TO FROM	World	Man'ment	Machines
World	----	ORDER	----
Man'ment	ACCEPT REJECT READY	----	ANN'MENT ASS'MENT PENALTY
Machines	----	BID W_START W_END	W_START

Fig. 2. Message types sent and received by the agents.

For evaluating the performance of the system, methodical experimentation, as presented in [15], has been started. In the first rounds of experiments handles have been collected that can influence the running characteristics of the above mechanism: among others, the management's time and cost thresholds of accepting orders, its options in filtering non-dominated bids, the machines' policy of schedule tree building and work selection. The first results are encouraging: for a given input stream random effects play very limited role and the handles work reasonably. Comparison of input streams with different parameters is the next job to accomplish.

Fig. 3. shows how the system processed an order stream of 8 jobs, 4 tasks each, on 3 machines with overlapping resources.

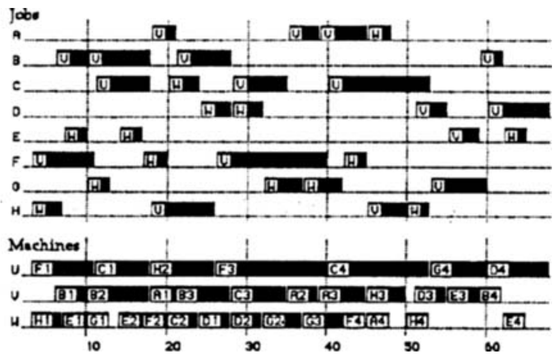

Fig. 3. The solution of a test problem

As a long term objective, we intend to develop an advisory system for selecting a proper market scheme for a given manufacturing system acting in a specific world.

6. Conclusions

Key features of this market-based control mechanism are as follows:

It supports *least commitment*, and, at the same time, *parallel commitment* problem solving: machines build up alternative futures in their schedule trees in a least commitment manner, while the Management's commitments are parallel since they refer to the same task. The latter feature is a novelty in cooperated negotiation, but it fades the chance of analyzing the system by means of control theory as suggested in [11].

The aggregate behavior of the system is *predictable*, and the protocol guarantees that each solution is complete and feasible; neither global constraint checking or simulation, nor tinkering the proposed solutions, so typical in distributed scheduling [3, 5, 14], are needed.

The system is *directable* through an incentive mechanism provided by the pricing scheme. By changing prices and penalties, the pattern of workload can be changed swiftly but smoothly. Due to local problem solving and parallel commitment, *conflicts* may occur. However, conflicts are resolved either by time or by the re-announcement mechanism of the market.

Though the system works by *reacting* to changes, *predictive* information is also available as far as management estimates the time and financial frames. The system is *open*: e.g., machines can be added or removed without any modification of the protocol.

7. Acknowledgment

This work has been supported in part by grants T-7341 and T-17442 of the National Research Foundation of Hungary (OTKA).

8. References

(1) Baker, A.D., 1992, Case study results with the market-driven contract net production planning and control system, Proc. of the AUTOFACT '92 Conf., Detroit, MI, SME, 31/17-31/35.

(2) Buzacott, J., 1995, A perspective on new paradigms in manufacturing, J. of Manufacturing Systems, 14(2): 118-125.

(3) Duffie, N.A., Prabhu, V.V., 1994, Real-time distributed scheduling of heterarchical manufacturing systems, J. of Manufacturing Systems, 13(2): 94-107.

(4) Glance, N.S., Huberman, B.A., 1994, Dynamics of social dilemmas, Scientific American, 270(3):76-81.

(5) Gou, L., Hasegawa, T., Luh, P.B., Tamura, S., Oblak, J.M., 1994, Holonic planning and scheduling for a robotic assembly testbed, Proc. of the 4th Int. Conf. on CIM and Automation Technology, IEEE, 142-149.

(6) Hatvany, J., 1985, Intelligence and cooperation in heterarchic manufacturing systems, Robotics and Computer-Integrated Manufacturing, 2(2):101-104.

(7) Hogg T., 1995, Social dilemmas in computational ecosystems, Proc. of the IJCAI-95 Conf., Morgan Kaufmann, 711-716.

(8) Iwata, K., Onosato, M., Koike, M., 1994, Random manufacturing systems: A new concept of manufacturing systems for production to order, Annals of the CIRP, 43(1):379-383.

(9) Lee, M.S., Rho, H.M., Kang, M.J., 1995, An evaluation system of order acceptability under consideration of machine loading in die manufacturing, Annals of the CIRP, 44(1):413-416.

(10) Maley, J.G., 1988, Managing the flow of intelligent parts, Robotics and Computer-Integrated Manufacturing, 4(3-4): 525-530.

(11) Prabhu, V.V., Duffie, N.A., 1995, Modelling and analysis of nonlinear dynamics in autonomous heterarchical manufacturing system control, Annals of the CIRP, 44(1): 425-428.

(12) Shaw, M.J., 1988, Dynamic scheduling in cellular manufacturing systems: A framework for networked decision making, J. of Manufacturing Systems, 7(2): 83-94.

(13) Smith, R.G., 1980, The contract net protocol: High-level communication and control in distributed problem solving, IEEE Trans. on Computers, C-29(12): 1104-1113.

(14) Sycara, K. Roth, S., Sadeh, N., Fox, M.S., 1991, Distributed constrained heuristic search, IEEE Trans. on Systems, Man, and Cybernetics, 21(6): 1446-1461.

(15) Valckenaers, P., Bonneville, F., Van Brussel, H., Bongaerts, L., Wyns, J., 1994, Results of the holonic manufacturing control system benchmark at KULeuven, Proc. of the 4th Int. Conf. on CIM and Automation Technology, IEEE, 128-133.

(16) Wellman, M.P., 1993, A market oriented programming environment and its application to distributed multicommodity flow problems, J. of Artificial Intelligence Research, 1:1-23.

(17) Wiendahl, H-P., Scholtissek, P., 1994, Management and control of complexity in manufacturing, Annals of the CIRP, 43(2): 533-540.