Survey of Agents and Multiagent's Systems Approach to the Modelling, Management and Control of Production Systems

Baltazár Frankovič, Ivana Budinská

Institute of Informatics, Slovak Academy of Sciences
Dubravska cesta 9, 845 07 Bratislava, Slovakia
frankovic@savba.sk; budinska@savba.sk

Abstract: The paper gives a survey about the MAS utilization for planning, scheduling, modelling, control and its execution; mainly applied on production systems. An application example considers a group of 4 products, production of each product is composed from 8 tasks. These tasks belong to one of 4 types of tasks and each of them may be executed by 2 equivalent machines.

Keywords: modelling, control, decision, multiagent system

1 Introduction

Planning, scheduling and control for an arbitrary system usually requires many important knowledge or information characterizing the behaviour of the system. On the basis of that information is possible to choose an appropriate method for designing of the available approach for the systems with different properties. Our goal is to describe the methods for the solution of in the title introduced elements. It is possible to denote that the Production Systems (PS), Supply Chain Management (SCM), Energetic systems (ES) and especially the Flexible Manufacturing Systems (FMS) belong to large systems (LS) which are defined as backward systems, where the primary inputs are the production requirements, the production conception, systems parameters etc., and the primary outputs are the final production, the quality of the products and the satisfaction of the customers, too. From this point of view it is necessary to deal with the development of their design and performance. Generally, in framework of LS, five main parts can be considered: planning, scheduling, simulation, control and execution. In this framework, planning process represents the periodical activity and aims to obtain the best scheduling of required tasks. Planning in manufacturing can be difficult because in planning, one must deal with detailed data, summary data, internal - external data, subjective information, and sometimes with no information at all.

The scheduling (which works together with the planning process) may be defined as the process of allocation of limited resources to production tasks on the basis of such information as for example: machine characteristics (as the resources of production process), production requirements, time of performance, production constraints, economical factors, etc.

The control system determines the sequences of control action for the resources used in the actual PS, SCM or other production and non production systems, too. The role of execution is to follow the performance of the system and to give backward information for the control system, which on the basis of this information creates the new available sequences of control actions. For the solution of so formulated problem it is possible to apply the rules of Multi Agent System (MAS). The solution of such problems is a subject of many publications [1][2][4][5][6].

2 The Significant Characteristics of MAS

Due to many advantages of the multi-agent system (MAS) and with respect to the complicated characteristics of the goal that we aim to achieve, the MAS approach seems to be the most feasible. There are some significant MAS characteristics that motivate us to choose the MAS approach:

- *Modularity*
- *Parallelism*
- *Flexibility*

Each intelligent agent can do reasoning about whom and when it has to cooperate with, in order to achieve the effective performance. Difficult questions are associated with the MAS approaches, e.g., which types of agents are needed, how many agents are optimal, what is a functionality of each agent, cooperation between agents, etc. We deal with all these problems during developing the system.

For this case we suppose that agents are technological or intelligent software entities with different properties, as for example:

- autonomy – agents act on their own to perform a task;
- proactivity – agents exhibit goal directed behavior and deliberation to solve a task;
- reactivity – agents can perceive the environments and react accordingly;
- social behavior – agents can communicate, and cooperate with other agents [3].

Generally the agents or coalitions of agents are suitable technology for developing the LS. Useful properties of agent technology include: the ability to perform complex calculation and analysis, interact with other LS automatically and quickly; scalability.

3 Agent-based Scheduling in Production System

The problem of scheduling production processes is discussed in a lot of papers. The scheduling process may be static or dynamic; it depends on requirements of the system. In this paper we analyse an approach to find a feasible schedule of all products, which may minimize a cost and satisfy all constrains related to each product. With the solution of this problem deal papers [1], [2], where the authors propose a mechanism to resolve a scheduling problem in real-time. In these papers they present a several mechanisms for negotiation and interfaces between high-level decision-making and lower-level scheduling and acting. In the work [3], the author deals with a problem of allocating and scheduling components of periodic tasks in distributed systems. In the paper [4], [5], the authors present an approach to agent control problem from a domain independent perspective. The scheduling problem is presented as choosing which tasks to perform and how to perform them to meet real-time constraints. Further they discuss a critical situation (time, cost) agents may produce, alternative plan selection and partial-order scheduling. The using MAS not only achieve better solution but also order more flexibly control in changing environment, quick reacting to change requirements. Project coordination in a multi-agent framework, where specification of MAS is in detail presented in [6]. The plan, scheduling, agent task, goal, information and communication between agents are explained too. We will focus essentially to agent negotiation to meet time conditions, a various aspects, which however are important for constructing plan like cost. In our case they have lower priorities as time condition.

4 General Description

We propose that in the system are several products necessary to produce and all their parameters and conditions are defined. In this case, each product may be considered as a set of tasks, which are operated in a certain type of machines, each product is limited by given deadline. Each task may be executed in a various equivalent machine but with different parameters therefore its duration and cost may be various according to where it is executed. Because a number of machines is finite and number of types of products may be large, it is necessary to find such sequence of all tasks that each machine's execution will be to optimal. It is clear that searching of optimal plan for execution one product can cause a conflict with plan of another products. To minimize their cost and to try to terminate before deadline, each product demands to use the best machine from its point of view for execution its tasks. Then it is needed to coordinate the agent's performance to find such plan, which may as most as possible satisfy each product's requirements.

In searching the optimal plan, 3 criteria are analysed:

- The all deadlines must be kept,
- the sum of all a cost must be minimal,
- if two previous criteria are filled up, the total time when the machines are used must be minimal.

Theoretically to find an optimal plan is a problem of searching through the space of all possible plans, each of which is a sequence of all tasks to execute, but the space may be a very large and it grows exponentially.

The set of all possible plans depends on count parameters (amount of products, number of operations). In case where an amount of products and number of operations are great, practical finding of optimal plan is not realizable.

Our idea presented in this paper is covered from 3 steps: the first step is that from any initial plan, which doesn't satisfy all given constrains, start to reconstruct this plan from bottom to top until finding such a plan, which can terminate before given deadlines; the second and the last step are that from this plan also continue to reconstruct to minimize costs and flow times of all machines. They may be cancelled at any moment if an obtained result is considered as feasible. In reconstructing and searching new plan, each product (we consider it to be one agent) can suppose one variant, which it considers as the optimal.

5 Characteristics of the MAS Basis Scheduling

The assumption is that there is a set of products and each of them has a set of tasks to terminate. A set of machines (resources) to execute these tasks are considered. Each possible plan appoints an order of operation when it may be executed and on which machine, for such plan there are 3 related parameters: the first is a time, when a last operation of each product is terminated (*Time_end*), the second is a cost of processing, when the plan is used (*Cost*) and the last is the total time when all machines are used (*Flow_time*). The plan with $Time\_end_i < t_i^0$ for every index i, where t_i^0 is *deadline* for product i, is called satisfied plan. The problem can be formulated as follows:

Definition 1: The problem of scheduling production processes is to find such plan δ^* from a set of satisfied plans (*Sat*) for which stands:

- $\nexists\ \delta \in Sat$ that $Cost(\delta) < Cost(\delta^*)$
- $\nexists\ \delta \in Sat$ that $Flow\_time(\delta) < Flow\_time(\delta^*)$.

Each task in production process has any set of predecessor and successor tasks (exception the first and last task has empty set of predecessor and successor respectively). These tasks need not to be independent of each other but may be related by a precedence relation $\Rightarrow$ that specifies whether these tasks must be executed one after another. $Task_{n,j} \Rightarrow Task_{n,k}$ it means $Task_{n,k}$ needs result of $task_{n,j}$. Two tasks from two different products also may be executed in the same type of machines and they can be executed in parallel. Let $T_{i,j}$ is a duration when *j-th* operation *i-th* product is spent in production process, $t_{i,j}$ is its start time. In the following definition necessary conditions for feasible plan are presented.

Definition 2: A plan is feasible if for all i,j,l,k the following conditions hold:

- $t_{i,j}+T_{i,j} < t_{l,k}$ if $Task_{i,j} \Rightarrow Task_{l,k}$
- if $[t_{i,j}, t_{i,j}+T_{i,j}] \cap [t_{l,k}, t_{l,k}+T_{l,k}] \# 0$ then $i \# l$

The parameter *Time_end* and *Flow_time* may be defined as:

Definition 3: for every $i \in <1,\ldots,n>, j$ and *plan* $\delta \in Sat$:

$$Time\_end_i = Te_i = \max_j \{t_{i,j}+T_{i,j}\}.$$

$$Flow\_time(\delta) = Flow(\delta) = \sum_{i,j} T_{i,j} ,$$

where $T_{i,j}$ depends on which of a set of equivalent machine is used to execute this task as mentioned above, each product is considered as one agent and it can access anytime to the machines. These agents have a common goal and coordinate to find a best plan for all. It is difficult to propose such mechanism for agents' negotiation, which can be applied in every situation, therefore our approach introduced in this paper may not find a globally optimal solution, but it can find a feasible sub-optimal solution from our point of view. The whole problem is presented as multi agent systems (MAS) and is defined as follows:

Definition 4: Multi agent production process (MAPP) is a structure

$$\{Ag, S, Cost, Time\_end, Flow\_time\},$$

where $Ag=\{1,2,\ldots,n\}$ is a set of agents (corresponding to amount of product),

S= set of all possible plans, $S \supseteq Sat$,

Cost, Time_end and *Flow_time* are criterion functions applied in each plan.

6 Negotiation Mechanism for MAPP

As mentioned in Section 5 each agent may prefer own goal before other and may propose such plan, which makes increasing these criterion functions that negotiation process may converge longer. Because of such problem it is better for the whole system that each agent in proposing own plan also considers what another agents must do and what are their constraints (for example deadlines). Another property is an amount of tasks what agents want to negotiate. As introduced in Section 5 a negotiation process can be stopped at any time if is found such plan $\delta \in Sat$, for all that in any case agents don't have to reconstruct whole plan if the initial plan is good, in such case is enough negotiate several last tasks. If the initial plan is far of optimal that whole plan must to be reconstructed, each agent must search in whole space of possible plans then it is better to choose another initial plan. A next point in this section is designated rule for choosing plan. It is assumed in any step are proposed *n* plans, for each plan is computed all its parameters (*Cost, Time_end* and *Flow_time*), from our point of view a most important criteria is satisfying deadline condition and next is *Cost* and last is *Flow_time*. Let plan proposed by agent *i* have parameter: $\{C_i, Te_{i1}, \ldots, Te_{in}, flow_i\}$. The agents must choose one of these plans, which they consider as best for all. There are several types of agents' negotiation.

6.1 A Volume of Negotiation

When negotiation begins, agents must choose how many tasks they want to separate, certainly this amount will increase until they cannot find appropriate plan. It is assumed each *i-th* agent want to separate m_i last tasks from its o_i tasks, because it supposes resorting these tasks may improve its all execution. From initial plan this agent can compute when each its task is started and ended. From definition 2 and 3 is easy to deduce a condition:

$$t_{i,j} \le t_i^0 - \sum_{k=j}^{o_i} T_{i,k} \quad \text{for every } j \in \langle 1, o_i \rangle. \tag{1}$$

A result deduced from this condition is: for every $i=\{1,2,\ldots,n\}$,

$$m_i \ge (o_i - j_{min}) \text{ where } j_{min} = \min j \mid t_{i,j} > t_i^0 - \sum_{k=j}^{o_i} T_{i,k} \ . \tag{2}$$

A next case when all agents agree with a plan, which satisfies deadline condition, to find a solution with smaller cost may realize as following.

	Part m-2	Part m-1	Part m

The whole plan is decide to m parts independent one after other, for example the last part includes all last tasks of each agent, part (m-1) includes all the second last tasks, etc. After deciding plan agents successively of bottom to top try to resort these tasks that belong to one part until find feasible plan with smaller cost, Let δ is initial plan, δ^1 is plan obtained of δ after reconstructing part m, notice δ^1 =recon (δ, m). In successive step plan δ^2 obtained by resorting part m and m-1, δ^2=recon(δ^1,m-1)…This process may be stopped if obtained result is not better as most as previous result or is enough good for our requirements. The proving of the next theory for simplicity is omitted.

Theory 1: A set of recont(δ^k,m-k) $\subset$ set of recont(δ^i,m-i) for $i<k$. and δ^k always satisfies deadline condition.

Similarly for searching min(Flow(δ)), therefore it is omitted.

7 Application Example

In our example a group of products, concretely 4 products composed from 8 tasks, these tasks also belong to one of 4 types of tasks and each of them may be executed in 2 equivalent machines is considered. Applying these theories discussed in Section 5 we designed such algorithm:

1. Initiative – give all parameters for tasks.
2. Choose any initial plan – may be arbitrary, at first we scheduled all tasks of agent 1 to respective machine, in next step agent 2, …etc.
3. Each agent propose an amount of tasks, which it want to resort in basis of theory in Section 6.
4. All agents cooperative to find a plan satisfied deadline condition. If is not possible that return to step 3 or 2.
5. The obtained plan is considered as new initial one, this plan is divided to m part independent as described in Section 4.3 – in our example the plan is divided to 4 parts, each of them includes 2 tasks of each agent.
6. Successively from bottom to top resort all tasks in each part of initial plan, after each step update initial plan as new obtained plan. This process is stopped if a new result is not better than previous one.
7. The chose plan after step 6 is stopping is considered as new initial plan and applying similarly as step 5.
8. Similarly as step 6, all agents search a new optimal plan, but one condition must be hold for new plan that is: new plan is feasible if (Cost (new plan)$\leq$Cost (initial plan)) $\wedge$ (Flow (new plan))$\leq$Flow (initial plan)).

For our example the step 3 and 4 terminates very quickly, but finding a plan with minimal cost is longer. Each agent uses a branch-and-bound algorithm to find own best plan, it don't search whole possible space but only such variant, which is desirable for its execution. In step 4 if agents don't successfully find a plan, they can return to step 2 and by applying (5) to test a condition of feasible plan introduced in definition 2. A process of finding a plan with min(Flow) is similar as finding a plan with min(Cost) but a condition shown in step 8 must always strictly hold to guarantee two previous filled up conditions. Applying this algorithm to our concrete example we obtained these results: From any accidental initial state after terminating step 4 was obtained such plan satisfied deadline conditions (plan 1), continue to execute steps 5 and 6 we obtained another plan with smaller cost (plan 2), but executing step 7, 8 did not bring a better or satisfied result. These results are shown in next Table 1.

Table 1
The results of concrete example

	Plan 1	Plan 2	Deadline (t^0_i | i=<1,..4>)
Te_1	29	28	35
Te_2	26	34	35
Te_3	29	29	35
Te_4	30	30	35
Total cost	432	404	
Flow time	108	108	

8 Design of Modelling and Control System -Decision System Created on the MAS Basis

Control of dynamic systems is a complex problem, which includes such sub-problems as decentralization, communication, global and local supervision, decision making, etc. Decentralized control is focused on local and global control problems and has to handle different classes of the decisions. A natural solution for control problems, for example in manufacturing systems is to follow a general motto 'think globally, act locally'. Decentralized supervisory control represents a sequence of control actions from the global supervisor to the local control operator, so called a local supervisor. The global supervisor collects and records information about events occurred in the system. The global supervisor is supposed to store a large knowledge about the entire system to compose a control strategy for each agent in the system. The control flow tends from the global supervisor to the local supervisors. A hierarchical decision process enables to use a hierarchical distribution of problems, where the recursive decision-making process on one level is composed by the following elements:

M_k – a decision process model on the k-level that contains a decision algorithm and the information and decision criterion.

$K=n$, n-1, … 1

1- represents the lowest level in the control hierarchy.

I_k –an information flow from M_k level to the M_{k-1} level

C_k – a scheduling information flow to the production components. Agents or a control flow

$S_{pk-1,k}$ – a back-loop information flow to the higher level

$H_{k,k}$ – an information flow among decision subsystems, models on the same level.

On the basis of this assumptions is possible to create the macro model of control system.

8.1 Macro-Model of Control

For the description macro model the following considerations are used:

D - a set of decision nodes

O - an open decision circuit, that means without the back-loop

FB - a closed decision circuit, contains the back-loop

U - a set of control actions, tasks, external events, and internal events

Tc - a set of direct controllable events

Tuc - a set of uncontrollable events

To - a set of observable events, (transitions), that are triggered inside subsystems, outside subsystems by another subsystem, respectively

Tuo - a set of unobservable transitions

A decision making period can be expressed by the following expression:

$$P_d = \frac{nodes \times levels}{number \quad of \quad level} = \frac{\sum_i d_{ik} \times \sum_k M_k}{k} \tag{3}$$

$$P_{d_{loc}} = \frac{number \quad of \quad nodes \quad in \quad one \quad level}{number \quad of \quad level} = \frac{\sum_i d_{ik}}{M_k} \tag{4}$$

This is a necessary but not sufficient condition for the system to change into a new state. An open circuit ***O*** of a control is considered. $k = n,\ ;\ n$ – number of the highest level.

The decision making period $Pd = 1$

Information I, that is necessary for the decision, tends directly from the k-1 level to the k-2 level and then from k-2 level to the k-3 level, etc.

The system changes to a new state only if the following condition is kept:

$$T_{uc,k-i} = f\left(I_k, H_{i,k-1}\right), \qquad i = 1,2,...,m \tag{5}$$

In the case of the controllable transition the following expression is satisfied: A control action for the entire system is the closed control circuit. There are the following equations for the transitions:

$$\begin{gathered} T_{c,k-i} = f\left(I_k, H_{i,k-1}, u_{k-i}, u_0\right) \\ u - \ a \ \ control \ \ action \\ u_0 - \ an \ \ initial \ \ control \ \ action \\ u_{k-i} = \begin{cases} enable \\ disable \end{cases} \end{gathered} \tag{6}$$

$$T_{uc,k-i} = f\left(I_k, H_{i,k-i}, S_{p,k-l}\right) \tag{7}$$

$$T_{c,k-i} = f\left(I_k, H_{i,k-1}, S_{p.k-1}, u_{k-1}, u_0\right) \tag{8}$$

The transitions in the entire system are the set of controllable and uncontrollable transitions:

$$T = T_c \cup T_{uc} \tag{9}$$

The observable transitions are realized transitions in the system. A decision process can be represented as an oriented graph

$$G^* = \left(IG, I, H, S_p, T_c\right) \tag{10}$$

The condition of the controllable transition is the information I_k, H_{k-1} .The decision algorithm represents a sequence of control actions for each agent in the system. Coordination among the agents in hierarchical systems are discussed in [9], [6]. An example of a special type of coordination among agents may be shown in the case of decision system.

9 Case Study of Market-Driven Model – Illustrative Example

The goal in a market-driven production system is to minimize differences between customers' demands and a real production on the other hand. The real production system is represented by a set of unreliable machines and workstations. The production control in a system with unreliable machines and random demands is dealt in [11]. An unreliability of machines is a stochastic variable. On the other hand customer demands are also random variable.

Let *r* - is a capacity of the production line. It depends on the machine is up or down. When the machine is down, the system can produce nothing and its capacity is 0. Average capacity depends on up and down periods of machines, which are exponentially distributed.

Let *d* represents average demands, which are homogeneous Poison distributed. For the writing a cost function we define a holding and backlog cost c+, c-

Then cost function is

$$c(x)=\begin{cases} c^{+}x & if\ x\geq 0 \\ -c^{-}x & otherwise \end{cases} \tag{11}$$

To ensure a stability of a market driven model, the average capacity of the production line has not be less than average demands.

$$\frac{rq_0}{q_1+q_0}\geq d \tag{12}$$

q_1, q_0 – duration of up and down periods

We can express a capacity and a demand gap:

$$\delta=\frac{rq_0}{q_1-q_0}-d \tag{13}$$

The aim is to minimize the operational cost by minimizing the production and demands gap over the infinite horizon, regardless of the inventory levels.

The threshold control policy [11]

$$u=\begin{cases} r & if\ x\leq n \\ 0 & otherwise \end{cases} \tag{14}$$

n – threshold level

The strategy for the production system is to determine optimal threshold level or so-called hedging point.

Let us describe a simple market-driven contract system. A bakery producing fresh croissants will be used to illustrate the theory introduced above.

There is a natural way how to decentralize the system.

There are three subsystems: production, delivery, and sale.

Each of these sub-systems can act independent and all of them will pursue the global goal.

All subsystems present an independent hierarchical system. A supervisor of the entire system is a management of all three subsystems. The global supervision of such a system requires negotiation of all three management. One of the optimal strategies for all of these sub-systems is combination of a market-driven order contract with a fixed order contract. It reduces a risk management, because it enables all of the three subjects to run on a sub-optimal base. For instance, a fixed order about 60% of the production capacity ensures the producer do not work far below the optimal production, the sale minimizes lose in spite of unsold products (croissants), and also distribution can organize its work in advance more effectively. These contracts represent an optimal threshold for the production system (a bakery). However there is uncertainty in a market-driven order with a tolerance about 40%. These orders are made day by day and enable all of the three subjects in the system to increase effectiveness and a profit.

The control strategy for the entire system, which includes three sub-systems, must be a mutual-beneficial solution. A control strategy for each of the three subjects must keep this global goal and then realize its strategy. The local strategy is subordinated to the global strategy. It is mandatory for all local supervisors to follow the global strategy.

The second level in the entire system is the supervision for all of the three subjects. The control problems generated by each of them are different. That means the control strategy for each subject is different. There isn't any interaction between agents on this level. We do not suppose a knowledge base for the entire system. There is just a unit of three knowledge bases on the highest level. The second level is final for the trade, if there is not any chain of shops. If there is a chain of markets, there is lower and lower levels until the last level represents single sale places. The control of such systems is beyond this paper. It includes many managers' activities. The second level in a distribution represents a management of a distribution company. The amount of levels depends on the size of the company and the lowest level represents single distribution devices (cars). The solution for such a systems is a classical traffic problem with the known number of cars and known number of customers modified for the flexible order. It means the company should be able to satisfy flexible orders up to 40% of the fixed order. This is the subject of negotiation on the highest level. Hierarchical problem on the production level: Input are daily orders, long term orders, real capacity of production line, disturbances, failed machines, material requirements, etc. Workstation level – capacity of machines, material requirements, disturbances. Machine level – strategy – to produce without disturbances. The scheduling problem of the multi-machine multi-product systems are dealt in details in [11] and [12]. There cannot be formulated a global criterion function for the entire system of the three subjects. On the highest level the supervisor just formulates constraint within the local control functions are formulated and will work. A negotiation protocol contains information obtained from the three subsystems. The negotiation on the highest level contains three agent negotiation. On the highest level it is cooperative negotiation, that means the agents look for distributed optimization. All conflicts are resolved in a cooperative form. Lower levels are represented by autonomous agents for each subsystem. The protocol could be either cooperative (distribution) or non cooperative (job shop scheduling). Strategies for the production on the highest level are as follows:

1 the production system accepts customer's order by the time of arriving. That means the system accept orders until capacity of production line is full. Another orders are refused.

2 the production system accept orders in spite of their values. Orders with lower value are refused if the capacity is full.

3 the production system accept all orders, but it reduces orders unitary, if the capacity is full.

Combination of these strategies can also give a good result.

An example: There is a production system with 4 machines. Capacity of machines and the cost of products produced on these machines are in Table 2. Table 3 contains the amounts of fixed orders and prices of products that shops pay per piece.

Table 2

A fictitious production system's parameters

	Capacity [pc/hour]	Cost pre piece
1	100	3
2	120	2
3	80	2
4	100	2

The aim is to maximize the cost function from the point of production system, which is the difference between production inventories and the profit (the sum of the prices of products paid by customers). Results of the illustrative example are depicted in Fig. 1.

Table 3
The fixed and flexible orders from customers

	Fixed orders[pc]	Price per piece	Flexible orders 1.day	2.day	3.day
1	800	3.00	400	300	400
2	400	3.50	400	300	400
3	400	4.00	300	300	400
4	600	3.00	200	300	400

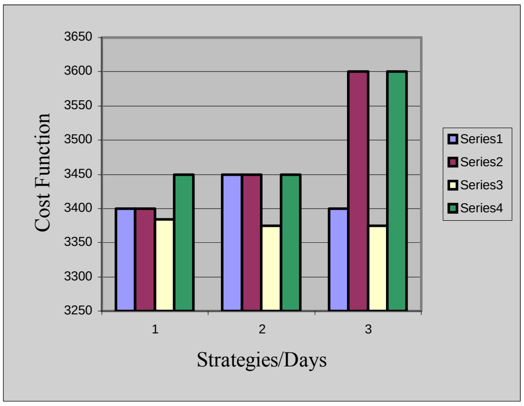

Figure 1
The cost function in dependence on the used strategy

Series 1 represents using the strategy 1, series 2 – strategy 2, series 3 – strategy 3, and series 4 is a combination of strategy 2 and 3. The combination of strategies 2 and 3 gives best results. That means the orders with the higher values are accepted and then the rest of orders with the same value are reduced unitary.

Conclusion and Discussion

The paper provides a basic introduction to the planning, scheduling and control of production systems using a hierarchical structure of a decision making (DM). A DM process created on the MAS basis may consider an intelligent and adaptive agent on the higher level and an autonomous agent in the lower level of the DM according to the system requirements. An example for the optimal or near optimal selection of the control strategy is presented here.

Acknowledgement

This work was supported by a Science and Technology Assistance Agency under the contract No. APVV LPP-0231-06; APVT -51-011602 VEGA 2/7101/27

References

[1] Peter J. G. Ramadge, W. Murray Wonham: The Control of Discrete Event Systems. Proceedings of the IEEE, Vol. 77, No. 1, pp. 81-98, January 1989

[2] Kai C. Wong, W. Murray Wonham: Modular Control and Coordination of Discrete-Event Systems. Discrete Event Dynamic Systems: Theory and Applications, 8, Kluwer Academic Publishers, Boston, pp. 247-297, 1998

[3] Karen Rudie, W.Murray Wonham: Think Globally, Act Locally: Decentralized Supervisory Control. IEEE Transactions of Automatic Control, Vol. 37, No. 11, pp. 1692-1708, November 1992

[4] P. Kozak, W. M. Wonham: Fully Decentralized Solution of Supervisory Control Problems. IEEE Transactions on Automatic Control, Vol. 40, No. 12, pp. 2094-2097, December 1995

[5] B. S. Gershwin: Hierarchical Flow Control, A Framework for Scheduling and Planning Discrete Events in Manufacturing System. Proc. of the IEEWE, Vol. 77, No. 1, pp. 195-209, 1989

[6] B. Frankovič, T. T. Dang: Scheduling of Production Using the Multi-Agent Approach by Hierarchical Structure. Proceedings of The Fourteenth International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems (IEA/AIE-2001), Budapest, Hungary, pp. 572-582, June 2001

[7] B. Frankovič, I. Budinská: Advantages and Disadvantages of Heuristic and Multiagent Approaches to the Solution of Scheduling Problem. Proc. of the IFAC Conference on Control System Design, Bratislava, pp. 372-378, 2000

[8] D. E. Wilkins, K. L. Myers: A Multi Agents Planning Architecture. Proceedings of the Fourth International Conference on AI Planning Systems, pp. 154-163, 1998

[9] Edmund H. Durfee, Thomas A. Montgomery: A Hierarchical Protocol for Coordinating Multiagent Behaviours. Proceedings of the Eighth National Conference on Artificial Intelligence, pp. 86-93, July 1990

[10] Y. Feng, H Yan: "Optimal Production Control in a Discrete Manufacturing System with Unreliable Machines and Random Demands." IEEE Transactions on Automatic Control, Vol. 45, No. 12, pp. 2280-2296, December 2000

[11] B. Frankovič, J. Fogel: Solution of Job Scheduling System. IFAC Conference MCPL Grenoble, 2000 Problem in Multi Part Production

[12] B. Frankovič, I. Budinská: Single and Multi-Machine Scheduling of Jobs in Production Systems. Tzafestas, S. G. (ed.): Advances in Manufacturing, Decision, Control and Information Technology. Springer London, pp. 25-36, 1999