Significant elements of the intelligent manufacturing systems

Frankovič Baltazár, Dang T.Tung, Budinská Ivana, Oravec Viktor

Dúbravská cesta 9, 845 07 Bratislava, frankovic@savba.sk

Abstract: The Production Systems (PS), Supply Chain Management (SCM), Energetic systems (ES) and especially the Flexible Manufacturing Systems (FMS) belong to large systems (LS) and may be considered as the intelligent manufacturing. Such systems are defined as a backward system, 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. The main elements of such systems are: planning, scheduling and control. The solution of the elements usually requires many important knowledge or information characterizing the behavior of the system. On the basis of that information and knowledge is possible to choose the appropriate intelligent methods. Our goal is to describe the methods for the solution of introduced elements. From this point of view it is necessary to deal with the development of their performance.

Keywords: Multi-Agent Systems (MAS), agents' coalition, logistics planning, scheduling, control, decision making

1 Introduction

The planning process represents the periodically activity and aims to obtaining the best scheduling of required technological task. Planning in manufacturing can be just difficult because in planning, one must deal with detailed data, summary data, internal - external data, subjective information and, sometimes, 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.

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.

The phases: planning, scheduling, control and execution may be more concurrent and it is necessary to consider the cycle time for each phase and also the whole production cycle [2], [4].

The solution of described problem requires to develop methodology (where it is necessary to suppose changes of customer or technological requirements) which is based on rapid, concurrent prototyping with frequent feedback to validate the requirements. Realization of such approach requires to use the intelligent methods as, for example parallel coordination, rapid collection of the actual information and also rapid communication. Then by this methodology, the actual state and the production dynamics can be obtained, as well. In the next parts we try to describe one special type of planning phase in LS (as, for example, PS, SCM, ES, FMS),denoted as the Logistics Planning System (LPS), the Scheduling Process (SchP), the Design of Control (DC) and execution of control action. The possible application may be, for example, the solution of multi-machine and multi part production systems created by different types of production lines (PL) which are in parallel configuration, where the jobs are structured in time. Each PL has its own production plan and their products are used to the completion of the final products. Because PL contains different devices (resources) in given configuration, according to the required technological process, it considers that some devices can serve for more than one PL. For the solution of so formulated problem is possible to apply the rules of Multi Agent System (MAS) .The solution of such problems is the subject of many publications [1][2][4][5][6] too.

This paper is organized as follows: Section 2 describes briefly the significant characteristics of MAS and agent behavior of each agent in the system, Section 3 deals with the formulation of LPS, Section 4 is devoted to the description of scheduling algorithms, Section 5 discusses the application of case-based reasoning and decision making for the design of control system, in Section 6 we discuss some related works for possible applications.

2 The significant characteristics of MAS

Multi-agent system (MAS) may be considered as the intelligent tool for the solution of the problems introduced above. The MAS approach seems to be most feasible. It respects the complicated characteristics of the goal that we aim to achieve. There are some significant reasons that motivate us to choose the MAS approach:

Modularity: each agent is an autonomous module and can work without interventions of the external world. Each agent can have different capabilities or functionalities and through cooperation the agents are able to achieve a variety of goals. From the practical point of view, producing a number of agents (e.g. software agents - programs) with different capabilities is more effective than creating one agent (e.g. a program), which is able to do everything. In addition, the MAS approach allows breaking the original problem solving down to a number of sub-problems with a manageable size. For example, many different agents could be used to assist a number of users at any time - there is a significant difference from the traditional centralized approach.

Parallelism: the MAS approach enables to work in parallel. A complicated problem could be solved in an acceptable time by using a number of agents, e.g., gathering information from various resources allocated in different places.

Flexibility: the MAS approach is able to react flexible to each change appeared in the environment. Through cooperation the agents can assist one the other to compensate the lack of capability or knowledge. They can share information or own capacity to resolve a newly appeared situation, if one agent is not able to do. Beside that, each intelligent agent can do reasoning about whom and when it has to cooperate with, in order to achieve the effective performance.

Of course, there are also some difficult questions associated with the MAS approach, 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 will have to successively deal with all these problems during developing this system, but in this paper we focus only on solving the problem: how the agent system can assist the user in designing the LPS, SchP and DC of any intelligent system.

3 Formulation of Logistic Planning System

Definition of Logistic Planning (LP) may be as follows: LP is a complex process, involving, collaboration among many distributed organizational and informational entities, locate and access required information, able to understand and analyze the information in order to form suitable deduction for the logistic plan in the form of a set of production task orders. LPS will utilize a restricted set of input options, such as the technological and time constraints, production tasks, production and customer requirements, objective functions, etc. Analysis of the logistics plan and available information is performed by the LPS to provide the logistics planner with information such as possible risks, any shortfalls in resources or suppliers, any perturbations, unsatisfied or relaxed constraints etc. For the solution of the introduced logistics planning problems agents' technology may be used [4][6][7].

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 a suitable technology for developing the LPS. They are able to model LS, to interact with other LS and communicate with users via users interface (UI). Useful properties of agent technology include: the ability to perform complex calculation and analysis, interact with other LS automatically and quickly; scalability - if the logistics system grows by the introduction of more LS, agents can be added to the LPS to represent these extra components or remove unused agents. This feature enables flexibility in managing LPS.

For the solution of LPS performance it is necessary to use an agent programming language that incorporates research into multi agent reasoning and reasoning under uncertainty. The first step for the realization of the LGS results is the scheduling of the given tasks.

4 Agent based scheduling in production system

The problem of scheduling production processes is discussed in a lot of papers. The scheduling process may be made in static or dynamic it depends on requirements of the system. In a case when it is enough a time to search appropriate plan that all possible plans may be examined, but in practice as most have to search for a short time therefore all requirements may be not filled up and in any case must to be contended with a plan, which is far of optimal plan. In this paper we analyse an approach to finding a feasible schedule of all products, which may minimize a cost and satisfy all constrains jointed with each product. With the solution of this problem deals in the [1], [2], where the authors propose a mechanism to resolve a scheduling problem in real-time case, in these papers they present a several mechanisms for negotiation and interfaces between high-level decision-making and lower-level scheduling and acting. They also discuss about a situation when a multiple methods excite for tasks and system needs to solve. In the work [3], the author deals a problem of allocating and scheduling components of periodic tasks in distributed systems, in this paper the authors also discuss about necessary conditions for static algorithms for such case. In the paper [4], [5], the authors present an approach to agent control problem from a domain independent perspective. Here the authors discuss about using agents to resolve scheduling problem, the scheduling problem is presented as choosing which tasks to perform and how to perform them to meet real-time constraints. Father they discuss about critical situation (time, cost) agents may produce, alternative plan selection and partial-order scheduling. The using multi-agent systems (MAS) to resolve scheduling problem is applied widely in a lot of works. The using MAS not only achieve better solution but also order more flexibly control in changing environment, quick reacting to change requirements. In [6] is presented project coordination in a multi-agent framework, where specification of MAS is in detail described. The plan, scheduling, agent task, goal, information and communication between agents are explained too. In this paper we will focus essentially to agent negotiation to meet time conditions, a various aspects, which however are important for constructing plan like cost, order etc. but in our case they have a fewer priorities as time condition. In next section will be introduced this problem.

4.1 General description

In our system are several products necessary to produce. 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 bounded 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 is executed. Otherwise could say that excite a set of methods for executing each task. Because a number of machines is finite and number of a type of products may be large, it is necessary to find a such sequence of all tasks to optimalize each machine's execution. It is easy to that in a searching optimal plan for execution one product it can cause a conflict with plan of another products, because to minimize their cost and try to terminate before deadline each product demands to use the best machine from its point of view for execution its tasks, therefore it is the best to consider each product to be one agent with all information and knowledge of all the machines and these agents try to coordinate between themselves to find such plan, which may as most as possible satisfy each product's requirements.

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 execution, but the space may be a very large and it grows with exponential speed. The set of all possible plans depends on count parameters (amount of products, number of operations), in a case where an amount of products and number of operations are great that practical finding 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 find a such plan, which can terminate before given deadlines, the second and last step are that from this plan also continue to reconstruct to minimize a cost and flow times of all machines. The second and thirst step may be cancelled at any moment if an obtained result may be considered as feasible. In reconstructing and searching new plan each product (we consider it to be one agent) can oneself suppose one variant, which considers as the best optimal, among these supposed variants all agents may choose one that as most meets all agent's requirements. Each agent attempts to fill up own goal, it may always suppose such variant, which satisfies as most itself therefore for successful choosing any variant must excite a rule that it clearly appoints to choose the most optimal one. The final plan may not be a globally optimal solution but at least a reasonable one.

4.2 Characteristics of the MAS basis scheduling

The assumption is made that it is a set of product (one agent) and each of them has a set of tasks to terminate, here is also a set of machines (resources) to executing these tasks. Each possible plan appoints an order of operation when it may be executed and in 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 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 following definition.

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

$\exists\ \delta \in Sat/$ that $Cost(\delta) < Cost(\delta^*)$

$\exists\ \delta \in Sat/$ 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 same type of machines and they can be executed parallel. Let notice $T_{i,j}$ 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 are presented necessary conditions for feasible plan.

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 $i \in <1,\dots,n>, j$ 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 follow:

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.

In next section we will focus to concept of negotiation among agents.

5 Negotiation mechanism for MAPP

In this section we begin with informal discussion of how the negotiation process may work. As mentioned in section 4.2 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 whole system that each agent in proposing own plan also considers what another agents must to do and what are their constraints (for example deadline). Another property is an amount of tasks what agents want to negotiate. As introduced in section 4.2, 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. The choosing initial plan is very important point, its quality can help reduce down a negotiation process to respectable time. 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, these agents not only work independently, they can also cooperative in smaller group to improve their plan before proposing. There are several types of agents' negotiation.

5.1 Application in example

In our example is considered 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. Applying these theories discussed in section 4 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 4.3
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)≤Cost (initial plan)) ∧ (Flow (new plan))≤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.

	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	

Table 1: The results of concrete example.

6 Design of modeling 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.
$K=n,\ n\text{-}1, \ldots\ 1$
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.

6.1 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 subsytems by another subsystem, respectively
Tuo - a set of unobservable transitions
A decision making period can be expressed by the following expression:

$$P_d = \frac{nodes \times levels}{number \quad of \quad level} = \frac{\sum_i d_{ik} \times \sum_k M_k}{k} \qquad (6)$$

$$P_{d_{loc}} = \frac{number \quad of \quad nodes \quad in \quad one \quad level}{number \quad of \quad level} = \frac{\sum_i d_{ik}}{M_k} \qquad (7)$$

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:

$$T_{uc,k-i} = f\left(I_k, H_{i,k-1}\right), \qquad i = 1,2,...,m \qquad (8)$$

In the case of the controllable transition the following expression is satisfied:

$$\begin{aligned} &T_{c,k-i} = f\left(I_k, H_{i,k-1}, u_{k-i}, u_0\right) \\ &u - \quad a \quad control \quad action \\ &u_0 - \quad an \quad initial \quad control \quad action \\ &u_{k-i} = \begin{cases} enable \\ disable \end{cases} \end{aligned} \qquad (9)$$

A control action for the entire system is the closed control circuit. There are the following equations for the transitions:

$$T_{uc,k-i} = f\left(I_k, H_{i,k-1}, S_{p,k-1}\right) \qquad (10)$$

$$T_{c,k-i} = f\left(I_k, H_{i,k-1}, S_{p.k-1}, u_{k-1}, u_0\right) \qquad (11)$$

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

$$T \; = T_c \cup T_{uc} \qquad (12)$$

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

$$G^* = \left(IB, I, U, H, S_p, T_c\right) \qquad (13)$$

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 [10], [7].
An example of a special type of coordination among agents may be shown in the case of decision system.

Conclusions

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.

Reference

[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 and 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, T. T. Dang and M. Filka: “Multi-agentové Hierarchické Decentralizované systémy.“ (in Slovak) AT&P Journal Vol. 11, pp. 42-45, 11/2000.

[8] B. Frankovič, I. Budinská: " Advantages and disadvantages of heuristic and multiagent approaches to the soltion of scheduling problem." Proc. of the IFAC Conference on Control System Design, Bratislava, pp. 372 – 378, 2000

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

[10] Edmund H. Durfee and Thomas A. Montgomery: "A hierarchical protocol for coordinating multiagent behaviors." Proceedings of the Eighth National Conference on Artificial Intelligence, pages 86--93, July 1990.

[11] 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

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

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

[14] A. Garvey, K. Decker and V. Lesser: A Negotiation-based Interface between a Real-time Scheduler and Decision-Maker. *Umass Computer Science Technical Report 94-08, 1994.*

[15] A. Garvey, M. Humphrey and V. Lesser: Task Interdependencies in Design-to-time Real-timeScheduling.