# Agent based approach to Mass-Oriented Production Planning: Case Study

Michal Pěchouček*, Martin Rehák*, Petr Charvát‡, Tomáš Vlček‡, Michal Kolář†
*Gerstner Laboratory, Czech Technical University in Prague, Czech Republic
‡CERTICON, a.s. Praha, Czech Republic, †gedas, sro. Mladá Boleslav
corresponding author: `pechouc@labe.felk.cvut.cz`

*Abstract*— **This paper discusses the potential of multi-agent planning techniques in the production planning domain, with special focus to mass-oriented production. The research presented in the paper has been centred around ExPlanTech – a specific implementation of production planning multi-agent system. Suitability of ExPlanTech for mass-oriented and project-driven manufacturing is discussed general in the paper. Applicability of multi-agent concepts is demonstrated on a multi-agent planning architecture and production planning case study at Skoda Auto Engine Plant.**

*Index Terms*— **production planning, multi-agent systems,**

## I. INTRODUCTION

In manufacturing the agent systems have been used in production control [1] and project-oriented production management [2]. While in manufacturing control the main added value provided by agent technology is in their control decentralisation, flexibility and robustness, agent-based project driven production planning provides mainly locally and globally balanced planning efficiency and very efficient replanning capability. In our article, we report on an extension of the agent planning techniques towards the mass-production domain. The experience we present was collected during the analysis, design and implementation of the actual planning system that is currently in the processes of deployment in SkodaAUTO engine production factory.

The typical problem of planning and scheduling in manufacturing is to allocate manufacturing activities to the available resources in the most optimal way so that manufacturing constrains are not violated. Manufacturing constrains are usually given by the manufacturing process specification and include a component list for a specific product, list of manufacturing activities, causal relations and list of available manufacturing resources that can be allocated to an activity. Besides manufacturing process specification, an important input to the planning process is also the current load and future commitments of the manufacturing resources.

The architecture we present is usable in both the project-oriented production and mass production context. In either of the domains we have based our solution on the ExPlanTech multi-agent architecture, that has been developed in part within the IST-1999-20171 European project [2]. In the former domain, the use of multi-agent planning becomes widespread and many prototypes and applications were developed. In the latter domain, the use of multi-agent planning systems faces more obstacles, as the emphasis put on plan efficiency is significantly higher and in most mass-production factories, multi-agent planning must compete with existing methods and prove its strengths.

In this article, we introduce the concept of AI planning and multi-agent planning techniques (see Section II) followed by specification of the different requirements imposed on the planning problem in manufacturing (see Section III). In Section IV we present a specific multi-agent architecture ExPlanTech and discuss its applicability in project-driven and mass-oriented production planning. Section V presents a specific case study of deployment of the ExPlanTech multi-agent solution in mass-oriented production at SkodaAUTO engine manufacturing site.

## II. FROM AI PLANNING TO MULTI-AGENT PLANNING

In this section we briefly explain the concept of multi-agent systems and the relevant agent-oriented and classical AI based planning techniques.

Classical artificial intelligence (AI) and operation research (OR) techniques are currently widely used in nowadays production planning systems. Besides classical linear programming (and related) approaches that have been widely used in the manufacturing problems where the problem can be described by a system of linear equations, there is a growing popularity in using various constrain satisfaction methods and constrain logic programming [3] for solving more complex planning problems. A modern and very straightforward approach to efficient planning and scheduling is based on the Theory of Constraint (TOC) [4], and integrates the classical concepts of the critical path and critical chain.

AI and OR provides various approaches to linear and non-linear planning based primarily on the informed methods of state-space search [5]. While little additional constrains need to be imposed on the planning problem, these methods tend to be rather computationally complex. Methods based on propositionalization of the planning problem have been proposed recently [6], [7].

The multi-agent systems provide different techniques and algorithms based on problem decomposition, task delegation and alternatives negotiation. Besides agent technologies provide development methods and programming infrastructures that can be used for implementing various systems for supporting non-trivial problem solving and planning problems. A multi-agent system is a collection of loosely coupled, autonomous agents. The agents are primarily computational, but hardware agents or human operators are often members of the multi-agent communities. Designing collective behavior of either cooperative or self-interested agents is the key engineering problem when developing a multi-agent system.

Multi-agent systems provide various techniques for planning the activities of distributed agents – *multi-agent planning*. Multi-agent planning usually deploys certain specific coordination technology. Centralised multi-agent planning is often based on the *black-board architecture* [8] or specific dedicated agents that can integrate various AI planning algorithms listed above. The most widely used approaches to distributed coordination are based on the *contract-net-protocol* [9], a seal-bid-auction type of a contraction mechanism or iterated contract-net-protocol (that corresponds to an English auction type of a contraction mechanism) [10]. The classical approach to multi-agent planning is the *Partial-Global-Planning* algorithm (PGP) that has been widely used in multi-agent systems [11]. PGP technique is a flexible approach to distributed planning and coordination that allows agents to dynamically reconsider their plans and commitments with the dynamically changing satiation. An efficient planing algorithm for cooperative multi-agent dynamic system is based on the use of large, structured Markov decision process (MDP) represented by a dynamic Bayesian Model [12].

## III. REQUIREMENTS FOR PRODUCTION PLANNING

In the domain of production planning, there is a whole set of different requirements that makes various planning algorithms hard to reuse. In this section, we will discuss selected requirements and properties of the problem that affect selection and design of the relevant multi-agent planning technique:

- **centrality:** is the environment distributed?
- **complexity:** how complex is the planning problem?
- **stability and replanning:** is the solution long-lived?

Let us discuss these properties in turn.

*Centrality:* We have to distinguish between the centrality of the problem and centrality of the solution. If the planning problem is centralized we assume that all the relevant planning data can be located centrally and can be analyzed by a centralized algorithm. If the planning problem is distributed, we expect that it is physically impossible (or unwanted) to locate all the actual data at one physical host (e.g. logistics planning). A problem can be central only to some extent (e.g. centrally available is only meta-information such as from where to where is the semi-product transported while the exact location of the truck is unknown). Obviously, centralized problems are more likely to be solved by centralized algorithms, while distributed problems would require more negotiation based approaches. However, a mix of central and distributed approach to central and distributed problems is possible.

*Complexity:* Production variability – both in the overall products/plan type ratio and in plan type/time unit ratio – is an important aspect to be considered when selecting a distributed planning algorithm. It is mistakenly assumed that distributed computation can solve elegantly NP-complete or exponentially complex tasks. Mere distribution into a fixed number of computational agents cannot and will not decrease computational requirements and there is no reason to believe that the solution will be provided dramatically earlier in comparison to the centralized solutions. Besides, sending a message between two agents usually requires more computational resources than passing a parameter when calling a method in a centralized system. However, multi-agent systems allow us to communicate imprecise, approximate knowledge. Agents can thus build approximate *acquaintance models* [13] that provide models of agents' imprecise mutual awareness. Knowledge stored in these models can be used as an important heuristic for finding a suboptimal solution in even very complex environments [14].

| Environment | Efficiency | Complexity | Stability |
|---|---|---|---|
| Process Production | **** | * | ** |
| Mass-oriented Production | *** | ** | ** |
| Project-oriented Production | ** | *** | *** |

TABLE I

KEY CHARACTERISTICS OF PLANNING PROBLEM IN DIFFERENT ENVIRONMENTS.

*Stability and Replanning:* As the plan is being executed, the inherent irregularities of the manufacturing process can cause deviations from the plan. Production control can autonomously handle many minor issues, creating a vast window of opportunity for multi-agent control [1] – each agent can manage its resources autonomously in order to fulfill the planned commitments. Some issues can be solved by negotiation between the subsequent processors in the manufacturing process, but some issues must be solved at a higher level as the local solution may interfere with the overall plan. On the other hand, handling of minor issues at a higher level of planning and control increases significantly the system complexity. The ExPlanTech architecture presented in section IV provides a flexible framework that adapts to these tradeoffs.

According to the Theory of Constraints (TOC), the output of most production facilities is defined by a bottleneck - a single fundamental constraint that limits the production capacity. Different types of manufacturing processes can imply different requirements on the planning, as shown in Table I and this section.

**Process production.** In continuous process plants (e.g. specialized chemical production processes) [15] require emphasis on real time control and appropriate incident handling, while the planning is mostly reduced to pure scheduling as the sequence of operations is typically predefined. Such systems are linear and have no or very limited storage capacities between processors. Therefore, the planning can be reduced to the selection of current processor capacity from the range values restricted by the maximum capacity of the bottleneck element for each element of time. The technology constrains – e.g. lead time, preparation and cleaning time and capacity limitation at startup or shutdown – needs to be naturally considered.

**Mass Oriented Production.** In the mass production context in the discrete manufacturing environment, the situation gets more complex as we touch the field for which the TOC was originally developed. The production systems typically have a single bottleneck, but the flexibility is increased as we may store the intermediary products in buffers between the processors. Range of products produced in one plant is typically wider and for each product, slightly different sequencing of operations in the plan may appear, even if the bulk of the plans remains similar due to the efficiency constraints. The differences between products may actually shift the production bottleneck along the line for various product mixes and this factor must be taken into account when we plan and schedule the production for individual processors in the sequence, together with the capacity of buffer storage and additional handling costs.

The main concern of planning and scheduling in both above cases is to ensure that the system capacity, defined by a well understood physical bottleneck is used efficiently and ensure that the production costs are globally minimized. Therefore, the main emphasis is put on scheduling and the planning is typically highly centralized.

**Project Oriented Production.** This is different in a project-driven production environment, where the assumptions laid down above no longer hold due to the increased variability of products and plans. Complex plans are executed by more versatile processors and implementation of tasks may be specific for each particular product. Such versatility makes the planning process much more complicated, as the single bottleneck that defines the plant output can be hardly found. The production capacity itself is defined by the number of projects completed rather than pieces produced - granularity of elementary lots on individual processors is typically higher. On the other hand, the profit margins in such processes tend to be higher than in the mass production and the emphasis is put on the feasibility of complex and diverse plans.

Therefore, the planning goal is no longer to optimize the global behavior defined by a system bottleneck, but to ensure that all processors are running at their maximum capacities and that the maximum number of projects is completed - we improve the efficiency by increasing the capacity and local optimization, rather than by smoothing the flows around the single well defined bottleneck. Our ability to plan the production is frequently an actual bottleneck that limits the output of the plant and restricts the efficiency. Centrality of the planning is low, and the systems tend to be more robust - small fluctuations in processing of individual lots don't propagate immediately as in linear processes, but can be absorbed by buffers.

In the following section, we will present a generic multi-agent architecture for production planning and control that we have developed for the project driven production environment, together with the specific modifications that enable the use of this technology in the mass-production context.

## IV. EXPLANTECH ARCHITECTURE

The ExPlanTech framework [2] adopts the ProPlanT multiagent architecture [16]. It contains an approximately fixed number of nontrivial agents, each providing different system functionality e.g. planning, simulation, and user access. We built ExPlanTech on top of the Java Agent Development Environment using the JADE platform[1]. For ExPlanTech the issue of interoperability and openness to 3rd part extensions is very important. This has been the main reason of the JADE platform deployment as it provides FIPA compliancy on various levels of agent integration. JADE efficiency has shown to be satisfactory in comparison to other competing environments available at the time of ExPlanTech development [17].

As the main goal of ExPlanTech was to be a general purpose framework, its organisational architecture does not comply with (but is open to) any production modeling methodology. The ExPlanTech system includes following types of agents:

**Planning Agents.** The core of the any ExPlanTech-based system is a community of appropriate planning agents (see Figure 1). A planning agent focuses on product configuration and quotation and creates production plans for individual orders. Production agent typically creates production plans by task decomposition and partial-order planning but may integrate various existing AI planning engines to handle different types of production - for example, linear programming, constraint logic programming, or genetic algorithm-based planning.

**Managing Agents.** An ExPlanTech like planning system may integrate one or many managing agents which may be in charge of detailed resource allocation and schedulling, provided that the planning agent does not work with the information about the agent current availability. Managing agents also take care of conflicts and manage replanning and plan reconfiguration.

**Resource Agents.** ExPlanTech features two types of agents for integrating or representing manufacturing resources (see Figure 1). These agents

- Integrate factory hardware and software systems (for example, creating a bridge to a material-resources-provision (MRP) system, or integrating PLC controllers)
- Simulate a specific machine, workshop, or department (for example, a computer numeric control machine or a computer aided design department).

Typically, many resource agents running in the system directly interact with a planning agent or a production managing agent and carry out the data gathering and specific data preprocessing.

Besides the baseline agents defined above, the architecture contains other specialized agents performing various system-wide functions: user interaction, meta-reasoning and extra-enterprise integration.

ExplanTech provides an architectural and technological framework for development of agent-based production planing systems. It has been to a various extent deployed in several industrial applications. In the following, we will only comment on its applications to project-driven and mass-oriented production planning.

---

[1] `http://jade.tilab.com`

### *A. Project-Driven Production Use Case*

The first ExPlanTech deployment was made in the pattern shop Modelarna Liaz, manufacturing dies, casts and molds for the car industry. The production is clearly project-driven, as a single (or very limited number of) product has been always manufactured from one design. Given the high replaceability of production elements, variability of the manufacturing processes, high average occupancy of the manufacturing elements and short delivery times, the resource allocation process has not been trivial. Given the fact that a project consisted on average of 5 - 10 production processes, where almost each process can be further divided into a high number of subprocesses we are talking about a state-space well over $10^{10}$.

In Modelarna Liaz there are **20 resource agents** integrating three 5-axis CNC machines, one wood workshop (staffed by 5 employees), one metal workshop (10 employees), twelve CAD designers and one finish workshop. The workshop agents have been directly linked with the data collection mechanisms running in the factory.

A **single planning agent** decomposes the production process into activities and task the resource agents. The difficulty is given by the fact that the optimal decomposition depends on the availability of the resource agents while their availability depends on the requested activity which is a result of the respective decomposition. The resource allocation process has been implemented by means of a *iterative subscription-based-protocol* (ISBC) [18]. The planning agent subscribes the resource agents for meta-representation of their occupancy (as a function of the amount tasked). The *Linear function approximation* has been used as an estimate of the resource agents occupancy. The planning agent uses this estimate for suggesting the most optimal project decomposition. The planning agent requests resource agents for the resulting amounts. It may happen that the estimate has not been accurate. If this is the case, the resource agent rejects to provide the resources as expected and sends the planning agent a counterproposal. This counterproposal is used for an update of the estimate and a new decomposition proposed by the planning agent. Within a very small number of iterations, the decomposition usually achieved is very close to optimum.

The deployment exercise has shown that the project oriented production is very suitable domain for deployment of multi-agent techniques. This is true mainly due to the following reasons:

- complex, multi-dimensional optimization requirements (in or case due-time and cost)
- on-line collection and integration of production data
- the physical model represented by the community of resource agents is non-permanent
- requirements for continuous re-planning, given by an arrival of higher priority projects or by changes on the manufacturing devices
- frequent interaction with several users
- quite often, there is a single project being planned at one time

These requirements suggest a multi-party negotiation based approach as an efficient alternative to classical monolithic planning.

### *B. Towards Mass Production Planning*

The same reasons that make multi-agent planning a perfect match for project driven plants severely affect the usability of purely multi-agent approach to planning in a mass-production environment[2]. The rigidity of the plant configuration and low variability of products reduce the potentials of negotiation. Moreover, decreasing granularity of tasks increases

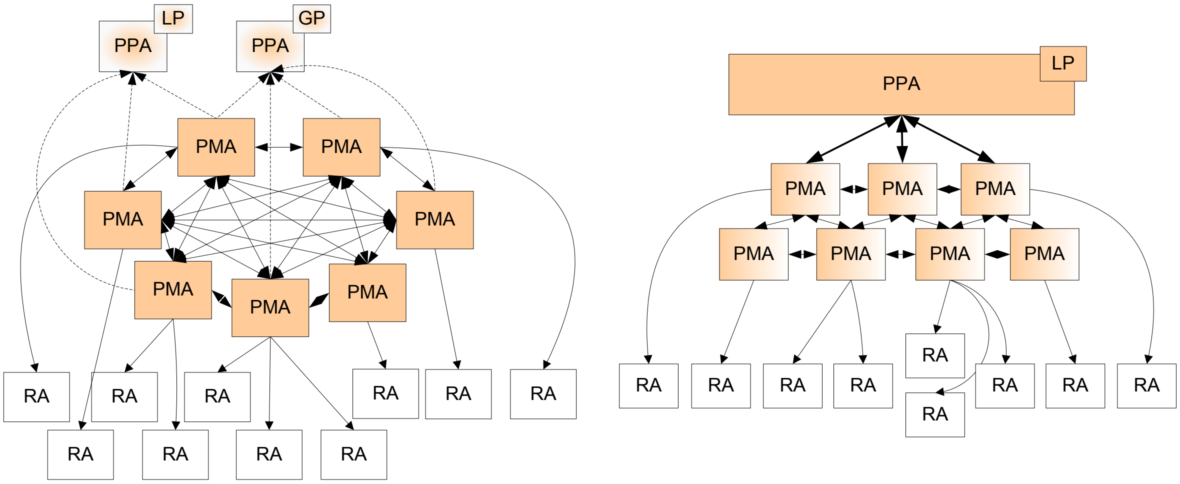


Fig. 1. ExPlanTech Architecture in project-driven environment (left) and in the mass-production environment (right). Mainly the dark agents are responsible for planning – PMA (stands for the managing agents) on the left and PPA (stands for planning agents) on the right, supported by semi-dark agents – PPA on the left and PMA on the right.

---

[2] In this context, we shall mention that we are considering traditional plants with rigid ordering of specialized tasks along one or more processors (chains). Multi-agent planning in plants with parallel processing capabilities provides better results. [19]

significantly the size of plan space. The utility differences between two slightly different plans are often negligible, making distributed approach impractical, as the negotiation costs (both in computational time and number of messages) become significant and improvements are only marginal for majority of modifications. Globally, we pass from negotiation about large, high-level tasks between autonomous and versatile processors to the negotiation between a small number of processors that concern units of pieces only.

The optimization criteria are also different (see Table II). In project driven production, we optimize locally for each processor and the global plan is just a combination of optimal local plans. This simple assumption does not hold anymore in the mass production case, as the cost of material and product handling between processors becomes significant - the criteria for optimization are therefore global (see section V-A), and difficult to ensure locally. In ExPlanTech architecture, we solve this issue by including the dedicated planning agent in the architecture to oversee such process. In Table II, we can see that the relative importance of various elements of ExPlanTech is different in the project-driven and mass production case. Instead of the distributed, negotiation-based approach to resource allocation, this duty was delegated to specialized planning agents that impose the plan to individual managing agents representing various processors. Unlike in the project driven case, the plan is actually imposed to managing agents whose autonomy is reduced – they merely oversee the execution of the imposed plan and report deviations to initiate replanning if necessary.

The overall planning performance is greatly determined by the actual planning method implemented by the planning agents and its adaptation to the current environment. On the other hand, the classical approach when we tune the planner during the system integration phase may be short-sighted. Long-term evolutions of the product mix may influence the planner results and reconfiguration may be necessary, imposing additional operation cost and greatly reducing the agility of the system. The same problem appears during reconstructions, holidays or other disruptions – the planner may be inadequate. Therefore, we need the planner to be as agile as the underlying multi-agent system, i.e. we need to close the loop and reconfigure the planner to reflect actual state of the production floor.

We suggest to use a meta-agent to tackle the problem. Meta-agent [16] is an independent monitoring and meta-reasoning agent that usually analyses behaviour of the agents and provides efficiency improvements. In our domain such an agent follows the changes of managing agents and possible plan deviations and can update the planner's model accordingly, therefore increasing its efficiency. As we may typically store the past production data, it is relatively easy to evaluate the new plan by comparison with the old one using the past performance and actual demand and status data. The plan with the best overall evaluation is then retained.

The evaluation criteria are also different from those applied in the project-driven production planning. As in our case, we typically try to achieve the following: *(i)* coverage of all planned deliveries, *(ii)* low stock volume for both finished products and material and *(iii)* high production uniformity, while respecting the *(iv)* additional constraints on material, tools and workforce availability.

| Property | Project Driven | Mass Production |
|---|---|---|
| Equipment Versatility | High | Low |
| Number of Requests | Low | High |
| Planning Variability | Low | High |
| Planning Granularity | Fine-Grained | Large Tasks |

TABLE II

DIFFERENCES BETWEEN PROJECT DRIVEN AND MASS PRODUCTION PLANNING.

## V. MULTI-LEVEL AGENT PLANNING IN EXPLANTECH: CASE STUDY IN SKODA AUTO ENGINE PLANT

We have successfully applied the elements of above-described ExPlanTech architecture to support the production planning in SkodaAUTO motor assembling plant. The project has been coordinated by gedas, s.r.o., while the Gerstner Laboratory at the Czech Technical University and CERTICON a.s. have contributed by a agent software architecture and a planning system prototype. The SkodaAUTO is in the process of deployment of the final robust planning system, implemented by Gedas for car engine manufacturing. This exemplifies high-volume production plant producing thousands of engines every day. A high variability exists in the types of motors to be manufactured. The planning system needed to provide the company management with detailed plans for a six weeks period, under the constraints presented in Section V-A.

While at the stage when we got involved in the project, the primary development target has been a stand-alone planning system, the further requirements were directed towards an open, interoperable and highly flexible system. It has been planned to allow integration with production monitoring and control tools, to allow real-time time re-planning in case of demand changes or production anomalies and also to allow easy and straightforward process reconfiguration of the planning tool at its run-time.

### A. Problem Statement

The automotive industry operates in high volumes and on very low margins, thus it focuses a lot of attention on process optimization. Such optimization can be specified by the following generic requirements, derived from the project specific requirements drawn by SkodaAUTO. For through motivation of these criteria, see [4].

- Minimize the stock through the production chain, thus decreasing the financial and storage costs.
- Maximize the production uniformity, to be able to use the industrial means in an efficient manner and to avoid overtime cost.
- Minimize the unnecessary handling of products between successive steps of the production process to further reduce human resources and other manipulation related costs.

- Allow the integration with production surveillance and management tools.
- Allow real-time or near to real-time re-planning in case of demand changes or production anomalies.
- Allow easy and straightforward process reconfiguration in the future.

It is interesting to note that the quantitative (first three) criteria listed above are contradictory and that a satisfactory optimum is their weighted, context dependent combination.

The respective factory contains three serially organized production lines, as shown in figure 2.

The factory production means can be described as three lines in a series, with two buffer stores and one main store used for final product storage before delivery and for the intermediary product storage as well. The intermediary product can be bought from outside or shipped directly to clients, as it may be a part of deliveries described below. Material store is not represented on the drawing, as the material is delivered to different positions on production lines when required. The demand is formalized as a matrix defining how many products of a given type shall be delivered on a given day.

### B. Multi-Level Planning

The factory and planning problem described above falls perfectly into the mass-production category: plans are rather similar, parallelism is limited to the production or use of intermediary products and the planning problem is practically reduced to scheduling. On the other hand, the emphasis we put on the scheduling efficiency is high, the state-space is enormous due to the high variety of products and the planning time-frame and the system must satisfy very strict criteria to ensure plant profitability.

In this case, a single, dedicated planning agent has a considerable advantage of global knowledge that allows it to use a heavyweight but efficient scheduling methods. On the other hand, a today's user will typically require a rapid process reconfiguration ability to follow process or manufacturing equipment evolutions and a close integration with on-line production surveillance and analysis tools. This is the area where agent-based distributed and adaptive systems have significant advantages [1]. As suggested above this deployment of the ExPlanTex architecture is based on a mix of these two approaches. We have divided the scheduling process into two distinctive phases, each of them with assigned scheduling tasks:

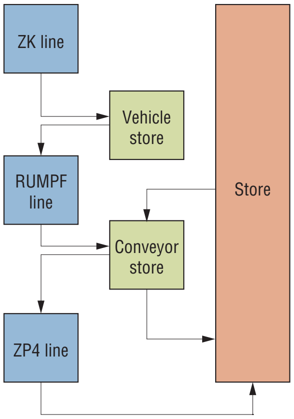


Fig. 2. Factory Layout

Fig. 3. Product - a completed car engine (installation of the engine in the vehicle is not carried out in the respective plant).

- **Long term planning** – Planning agent is responsible for the fulfillment of the first two requirements, the minimum stock and maximum production uniformity, together with ensuring that the deliveries are feasible. The output of this planning stage is the size of lots to produce on different lines during the given day.
- **Short term scheduling** – Managing agents process the output of the high level scheduling and organize the work on their lines during the appropriate time period by distributing the lots of products into appropriate timeslots. In this planning phase, we handle the production continuity, ensuring that the line 2 immediately consumes the product assembled on the line 1 and that the same applies for the lines 2 and 3.

Production surveillance may be connected to this process for purposes of dynamic rescheduling. Extensions of Multi-level planning are discussed in Section VII.

### C. Long Term Planning

The high level scheduling is managed by one or more dedicated planning agents. Each of these agents encapsulates one or more planning methods and uses them to create and maintain the high-level plan. When there is only a single planning agent, its plan is retained and used to manage the future production. When more than one plan is available, agents must determine which plan satisfies the criteria (stated in pervious section) best and this plan is retained.

The goal of long term planning is to allocate the appropriate number of product groups to be manufactured on specific processors. The fitness function to be optimised here minimises the production variability and the amount of material in the manufacturing process (especially in the stock).

As the various production capacity constrains (as well as the fitness function) can be represented by a system of linear equations. The planning agent has been implemented by mean of multi-stage linear programming. However, application of linear programming to our planning problem was not straightforward, because the problem as specified is not entirely linear. We had to resolve the following issues:

1) the lot-size has to be an integer,
2) the production uniformity relation is non-linear,
3) the lot-size has to be either zero or more than the minimum value.

The integer values of lot sizes are an issue that is easy to resolve. Either we can use integer extension of the standard LP algorithm, which is NP-hard and complicates the solution, or we can simply ignore this issue and round the results of the LP algorithm. The rounding error caused by this approach is (in most cases, as well as in our case) insignificant compared to total number of products.

Non-linearity of the production uniformity relation is an issue that is much harder to resolve. We have opted for an alternative approach that modifies the conditions of the original model by requesting the production not to divert from the average required production value by more than specific percentage. The average load value used in the relation to determine the boundary values can be calculated per processor, per processor group or for the whole plant, depending on client preferences.

The third problem, the minimum lot size issue was also solved by iterated runs of the algorithm. If the product of the first run of the algorithm was produced on a given day in a quantity larger than a predefined ratio of the minimum lot size, we require the minimum production of this product in a given day to be at least the minimum lot size.

Use of the linear programming under nominal conditions guarantees that we can find the optimal solution in a polynomial time. However, our case is far from being nominal - minimum lot size and low variability requirements make the problem non-linear and impose additional requirements on the solution. Therefore, we are not always sure to find an answer. On the other hand, in case of failure, linear programming allows identification of the constraint that caused the failure, thus facilitating human intervention.

As an alternative planner, we have developed an approach based on genetic algorithms [20]. We have defined one member of the population as one possible plan and we have defined a fitness function reflecting all of the high-level scheduling criteria described above, and self-correcting breeding and mutation operators to ensure that the bulk of the population respects the basic criteria for the plan validity. This approach was necessary to obtain the results in a reasonable amount of time – without these fine-tuned operators, it was almost impossible to find a solution respecting the restrictions.

In contrast to the linear programming approach, there was no need to keep the criteria linear and the development of teh fitness function was straightforward. On the other hand, the problem was heavily constrained, and we were forced to implement task-specific breeding operator to filter-out the unacceptable plans, as mentioned above. With the application of this operator, we were able to find a stable, acceptable plan after about 15 min. of computation on a standard PC, with the algorithm implemented in C++.

On the other hand, the linear programming solver performed comparably in quality and despite the size of the problem (several thousands of relations/columns, depending on the number of working days and product types), we were able to find the solution in less than 1 second, with 1 LP iteration taking a mere fraction of second to compute.

Therefore, our choice for the final implementation was an easy one and we have selected linear programming. The main factors beyond this choice were its speed, robustness and an ability to detect the constraint preventing us from achieving our goal. The fact that some of the criteria (production variability/minimum lot size) had to be enforced in a non-standard ways was considered as insignificant by the client.

### *D. Short Term Scheduling*

High level plan prepared by the planning agents provides enough details for material acquisition and resource planing for a six weeks period. However, in order to execute the plan in an optimal manner, we need to make it much more detailed; instead of simply providing the number of products of a given type to be produced in a given day, we must prepare the actual sequence of production batches and their allocation and scheduling to various machines/processors in the plant. At this planning stage, we must fulfill the requirements concerning the production fluidity, preferred ordering of different types of products and minimal handling – production of subsequent elements in the chain shall be synchronized whenever possible, as the synchronization minimizes the handling costs.

Therefore, we see that we no longer seek the overall plant optimization that was already achieved in the high-level planning stage, but we seek to locally optimize the production switching and handling costs between the processors on the line. Each management agent was assigned exact quantity of each product to produce and it must negotiate with the other agents to optimally manage their processing resources while achieving this goal.

In our model, we associate a cost to switching between two types of products on a line. This cost is relatively low for products that share a major part of their components, but grows with increasing differences between products.

We also associate a similar cost to handling of products between successive processors. Even if it is possible (and sometimes necessary) to process different batches on successive machines, this operation causes supplementary costs due to the material handling and storage. Therefore, we prefer the plans where the product is produced at once, i.e. where each machine directly processes the output of its predecessor in the chain.

*Single-Agent Planning:* For a single agent, that manages one processor (e.g. one of the three lines, conveyor or store), we may describe the ordering of different products it processes as an oriented graph, where the nodes represent the lots and the evaluations of edges connecting them represent the cost of succession of these particular lots in the order determined by the orientation of edges.

To introduce the handling costs into the model, we may rely on virtual payments to handling agents that bring the product to and from the store – the products acquired directly on the line will be therefore cheaper.

The lots scheduled for one day are connected in both directions, as any of them may precede another, while the connections between successive days are only from the past to the future. When we establish such graph for all the lots during the planning period, we may note that finding an optimum order is equivalent to finding a Hamiltonian path (a path passing through all the nodes) through this graph, connecting the first and the last lot. Such task is NP complete (and is only a part of our planning problem) and there is no trivial way of finding an optimal solution.

Each agent pre-orders the batches in an optimal manner and uses this simple ordering as an input for future negotiations with other agents. As stated above, pre-ordering is not trivial. However, the frequent changes of delivery orders, together with non-uniformities of the production process make the complete and uninterrupted execution of the plan highly unlikely. Therefore, we have opted for the use of the greedy algorithm, which seeks the immediate optimum and always selects the batch with locally cheapest transition – expecting that future gain from accepting the immediately suboptimal decision is highly unlikely to be collected anyway.

*Inter-Agent Planning:* All the managing agents use the utility function above to fulfill their individual goals. However, the goals of individuals may be conflicting and plan elaboration requires negotiation between individual agents. The aim of the negotiation is to produce a detailed plan, accepted by all agents and being a close to optimum as possible (See Fig. 4).

In the negotiation process, we reuse the inputs from the high-level planning agent and each agent's utility function. The mechanism we apply is based on a simple assumption – the key players, with the greatest power in the negotiations, are those using their resources most. This also corresponds to the TOC principles, as this machine is equivalent to the bottleneck in the problem defined by the high-level plan and production constraints. Intuitively, we may also note that the free capacity of the other machines allows them to better adapt to a locally suboptimal plan. This shall ensure the feasibility of fulfillment of the goals specified by the high-level scheduler.

We propose a specific negotiation structure that is depicted in Figure 4. The structure consist of the mutually interlinked activities of the agents that are explained in the following list:

1) All agents receive their goals prescribed by the high-level plan.
2) Each agent computes the ratio $r$ of the necessary/nominal load and prepares locally-optimal order of batches. Agents exchange the ratio $r$ (via peer-to-peer broadcasts or through middle agent).
3) The agent with highest $r$ – $A_0$ starts the negotiation by submitting its plan to the other agents in the chainin the direction towards the second most charged agent $A_1$.
4) The negotiation starts by the submission of the requests from $A_0$ towards the immediate producer/supplier, who analyzes the plan, compares it with its own plan and determines whether it is reasonably close to local optimum.
   - If the plan is acceptable, it passes the requirements to the next agent(s) in the chain.
   - If the plan is not acceptable the agent rejects the plan and proposes plan improvements The improvement proposal can be *(i)* accepted without changes, *(ii)* refused, *(iii)* answered with compromise version or *(iv)* can result in re-negotiation further in the chain, towards the original planning source.
5) The process is repeated until we reach the end of the production chain in the original direction of planning.
6) After this, the same process (e.g. starting at step 3) is initiated starting form the agent $A_0$ selected at the step 3 in the in the opposite direction.
7) Secondary plan branches (not present on the critical path) are calculated and their final plan is determined.

It shall be noted that only a limited number of negotiation attempts is allowed between any two adjacent agents to avoid infinite loops. This limitation makes the algorithm an O(n), where n is a number of agents. On the other hand, such limitation can cause the algorithm to miss a better solution or it may prohibit it from finding the solution altogether. In such situation, we re-start the negotiation with a new initial agent, selecting the agent where the incompatibility was detected and negotiating towards the bottleneck first. When the new attempt fails as well, we repeat the high level planning with more restrictive parameters, such as increased intermediary stock level to address the inconsistency. In the general case, it is impossible to prove that the algorithm will always converge towards a valid solution, but for most practical applications, the adjustments between both planning levels can address most issues.

Note that the above described algorithm outline assumes that *(i)* the agents have social knowledge about the others (providers/suppliers), that *(ii)* the environment is collaborative and that *(iii)* the plant ordering is mostly linear. These assumptions were respected in our case and are reasonable for most of mass-production plants. Very specific cases of agent are agents that are responsible for possible branching – such as vehicle store/store and conveyor/store agents in our case. These agents must be careful to prioritize the planning on the critical path (branch), and prepare the requests for the secondary branches once the main branch is planned. This approach follows the classic TOC approach, where we define the critical path and the critical chain concepts[3].

*Holonic approach:* Yet another mechanism implementing daily production lots ordering can based on an alternative approach to agent modelling in the manufacturing domain. While in the described, ExPlanTech like architectures the agents rather permanent and represent manufacturing devices, the PROSA-like [21] architecture suggest representing also a lot (in some applications even a singular product) as an agent. These agents, in PROSA denoted as *order agents*, requests the *resource agents* for allocating an appropriate time slot on the particular

---

[3] However, in the algorithm presented above, we don't consider the case when a single resource is used on multiple branches (critical chain). We only consider the cases where the critical path and critical chain are identical.

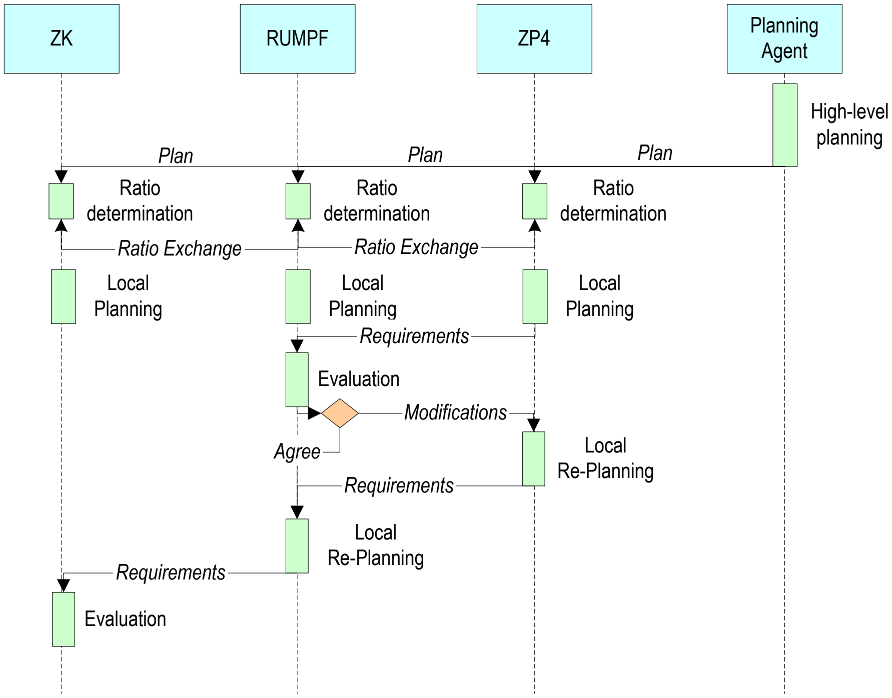


Fig. 4. Short Term Scheduling Negotiation Architecture.

manufacturing processors, represented by the given resource agent.

Optimization can be ensured again by an agents' utility function, similarly to above. The resource agent does not process all the requests sequentially, while it collects all the requirements from all the interested order agents and suggest resource allocation according to orders priorities and delivery due dates. Holonic approach have not been deployed in our system.

### *E. Production surveillance and dynamic re-planning*

In a real environment, we can not expect the plans to be executed smoothly and flawlessly. Therefore, we shall extend the planning system to *(i)* prepare plans that are robust to frequently occurring problems and *(ii)* dynamically integrates production surveillance to automatically replan in real-time. Today, such adjustments are decided by human managers based on their observations of the production process and their experience, even if automatic solutions start to appear [1]. An agent system shall have a similar functionality – technical issues are at first detected by a resource agent and the managing agents are notified when the problem limits the capacity, while the deviations from the plan (caused e.g. by material unavailability or quality problems) are detected by managing agents directly.

When a managing agent detects a problem in its zone of responsibility, it tries to resolve the problem locally without disturbing the operations of adjacent agents.

We propose that the processor planning agents shall be also used for online process surveillance and that the information gathered by these agents may be used both to increase the experience of agents and to react immediately to current situation, using the agent experience [19] and [13]. This approach would allow us to eliminate many false alarms, connected with slow production start or short-time material inaccessibility, but can sooner detect potential major problems, especially by looping back the information form quality assurance stations.

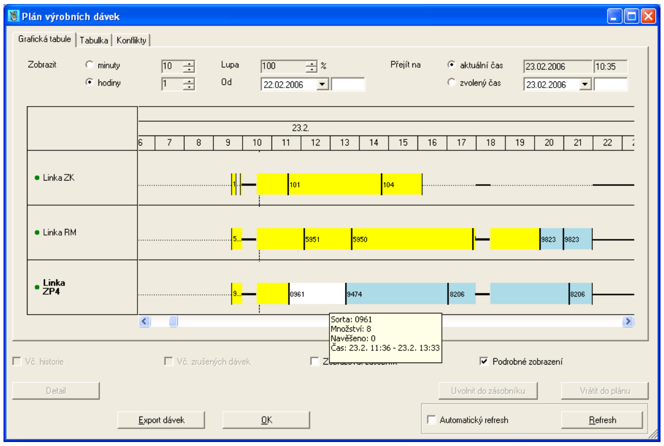
## VI. IMPLEMENTATION STATUS

The described production planning system in Skoda Auto is an important part of a modular MES system (Manufacturing Execution System), which is designed to cover in successive steps all of 11 areas of functional model of MES, (see www.messa.org).

Besides necessary interfaces between the company ERP systems, the developed MES system contains modules supporting quality management, production surveillance, production scheduling and long term planning. All these components are fully tested and being introduced to real manufacturing process (Autumn 2005).

For the implementation of our current long term planner, a free third party LP solver was used, together with communication and data transformation wrapper. The whole scheduling takes less then 1 second on standard PC (with 28 days, 50 products and 3 machines). This completely satisfies the performance requirements. The short-term scheduler has been fully developed at Gedas, s.r.o.

Due to sophisticated design, usage of progressive technologies, robust IT architecture and good collaboration between the customer and the implementation team is the implementation assessed from the customer side as successful. Targets of the project were fulfilled. The important side effect of our



Zobrazit ○ minuty 10
● hodiny 1
Lupa 100 %
Od 22.02.2006
Přejít na ● aktuální čas 23.02.2006 10:35
○ zvolený čas 23.02.2006

Sorta: 0961
Množství: 8
Navěšeno: 0
Čas: 23.2. 11:36 - 23.2. 13:33

☐ Vč. historie
☐ Vč. zrušených dávek
☐ Z
☑ Podrobné zobrazení
☐ Automatický refresh

Fig. 5. Graphical user interface of the developed system.

project was elimination of an excessive work items that did not provide any added value.

## VII. CONCLUSIONS

The work we describe is based on an industrial application project where Gerstner laboratory and CertiCon, a.s. provided research results and agent-based planning architecture, while Gedas, s.r.o. has carried out industrial deployment of the technology. The major research contribution presented in this article is in practical realization of planning architecture that separates high-level and low-level scheduling, enabling us to benefit both of the global view of the dedicated scheduling component and flexibility, reactivity and potential learning ability of agent based systems. We've successfully integrated linear programming methods into the project solution and demonstrated the complementarity between this classical approach and dynamic agent systems.

Classical AI planning methods with the agent component cannot have been used in our in the presented case due to nonlinearity and high computational complexity of the problem. Separation of the planning levels reduce the computational complexity and allows to use the classical AI planning techniques for the higher level, longer-term planning while the low-level negotiation is feasible in the reduced space.

The approach present does not have the ambition to find an optimal production plan (which would mean solving an exponentially complex computational problem in a 'reasonable' time). Instead it encodes knowledge about the logics of the manufacturing problem as a 'heuristics' into a multi-agent production model that is used for short term, negotiation based scheduling.

Unlike other deployments of agent system in manufacturing (e.g. [1], [2]) the presented application case does not exploit the aspect of physical distribution of computation process and data. Similarly to [2], it provides an increased robustness of plans. However the key contribution of agent technologies in mass-production oriented domain is in agents capability to simulate an unfeasible production plan (provided by efficient long term planning algorithms) detect conflicts, negotiate their local objections and provide solution.

## ACKNOWLEDGMENT

This work was supported in part by the Ministry of Education, Youth and Sports of the Czech Republic under the grant no. MSM6840770013 and by the EC project I*PROMS: The Network of Excellence for Innovative Production Machines and Systems, (No. FP6-500273).

## REFERENCES

[1] S. Bussman and K. Schild, "An agent-based approach to the control of flexible production systems," in *Proc. of the 8th IEEE Int. Conf. on Emergent Technologies and Factory Automation (ETFA 2001)*, vol. 2, 2001, pp. 481–488.

[2] M. Pěchouček, A. Říha, J. Vokřínek, V. Mařík, and V. Pražma, "Explantech: applying multi-agent systems in production planning," *International Journal of Production Research*, vol. 40, no. 15, pp. 3681–3692, 2002.

[3] R. Barták, "Dynamic constraint models for planning and scheduling problems," in *Selected papers from the Joint ERCIM/Compulog Net Workshop on New Trends in Contraints*. London, UK: Springer-Verlag, 2000, pp. 237–255.

[4] E. M. Goldratt, *The Theory of Constraints*. N.Y.: North River Press, Croton-on-Hudson, N.Y., 1990.

[5] F. Giunchiglia and L. Spalazzi, "Intelligent planning: a decomposition and abstraction based approach to classical planning," *Artif. Intell.*, vol. 111, no. 1-2, pp. 329–338, 1999.

[6] I. Miguel, P. Jarvis, and Q. Shen, "Flexible graphplan," in *Proceedings of the Fourteenth European Conference on Artificial Intelligence*, W. Horn, Ed., 2000, pp. 506–510.

[7] C. P. Gomes, "On the intersection of ai and or," *Knowl. Eng. Rev.*, vol. 16, no. 1, pp. 1–4, 2001.

[8] C. R. E. Szczerbicki, "Blackboard approach in design planning for concurrent engineering environment," *Cybernetics & Systems,*, vol. 29, no. 7, pp. 729 – 750, Taylor & Francis 1998.

[9] R. G. Smith, "The contract net protocol: High level communication and control in a distributed problem solver," *In IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, 1980.

[10] T. Sandholm, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Cambridge, MA.: MIT Press, 1999, ch. Distributed Rational Decision Making, pp. 201–258.

[11] K. Decker and V. Lesser, "Generalizing the Partial Global Planning Algorithm," *International Journal on Intelligent Cooperative Information Systems*, vol. 1, no. 2, pp. 319–346, June 1992. [Online]. Available: http://mas.cs.umass.edu/paper/34

[12] D. K. C. Guestrin and R. Parr., "Multiagent planning with factored mdps," in *In Advances in Neural Information Processing Systems (NIPS-14),*, Vancouver, Canada, December 2001, pp. 1523 – 1530.

[13] M. Pěchouček, V. Mařík, and O. Štěpánková, "Role of acquaintance models in agent-based production planning systems," in *Cooperative Infromation Agents IV - LNAI No. 1860*, M. Klusch and L. Kerschberg, Eds. Heidelberg: Springer-Verlag, Heidelberg, July 2000, pp. 179–190.

[14] O. Lerch, "Effcient contraction mechanisms for virtual enterprises operation," Master's thesis, Czech Technical University, 2005.

[15] M. Rollo, P. Novák, J. Kubalík, and M. Pěchouček, "Alarm root cause detection system." in *In Emerging Solutions for Future Manufacturing Systems*, New York: Springer,, 2001, pp. 109 – 116.

[16] V. Mařík, M. Pěchouček, O. Štěpánková, and J. Lažanský, "Proplant: Multi-agent system for production planning," *Applied Artificial Intelligence*, vol. 14, no. 7, pp. 727–762, 2000.

[17] P. Vrba, "Java-based agent platform evaluation," in *Holonic and Multi-Agent Systems for Manufacturing*, ser. LNAI, Mařík, McFarlane, and Valckenaers, Eds., no. 2744. Springer-Verlag, Heidelberg, June 2003, pp. 47–58.

[18] M. Pěchouček, "Role of subscription based contraction in agent based production planning," in *EUMAS*, 2003.

[19] V. Marik and D. McFarlane, "Industrial adoption of agent-based technologies," *IEEE Intelligent Systems*, vol. 20, no. 1, pp. 27–35, January/February 2005.

[20] J. Kubalik, L. Rothkrantz, and J. Lazansky, "Genetic algorithm with limited convergence," in *Information Processing with Evolutionary Algorithms: From Industrial Applications to Academic Speculations*, M. G. et al., Ed. JCIS/Assoc. for Intelligent Machinery, 2005, p. 216235.

[21] H. V. Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: Prosa," *Computers in Industry,*, vol. 37, no. 1, p. 255 274, 1998.