PROJECT-ORIENTED APPROACH TO PRODUCTION PLANNING AND SCHEDULING IN MAKE-TO-ORDER MANUFACTURING

PÉTER EGRI, ANDRÁS KOVÁCS, ANDRÁS MÁRKUS, JÓZSEF VÁNCZA
Computer and Automation Research Institute
Hungarian Academy of Sciences
H-1111 Budapest, Kende u 13-17
HUNGARY
{egri,akovacs,markus,vancza}@sztaki.hu

[Received December 2004 and accepted January 2005]

Abstract. In this paper, we present a unified framework for production planning and scheduling in make-to-order manufacturing. Both planning and scheduling problems are captured as resource-constrained project scheduling problems. The actual models and solution techniques are different at the two levels. Hence, we also suggest an aggregation method that provides the connection between planning and scheduling. The viability of the approach is demonstrated by some experimental results on large-scale industrial problem instances.

Keywords: Production planning, scheduling, aggregation

1. INTRODUCTION

Production planning and *scheduling* (PPS) match future production load and capacities by determining the flow of materials and the use of resources, over various horizons and on different levels of detail. Albeit planning and scheduling problems have their own, specific timescale, resource and activity model granularity as well as optimization criteria, the two levels of PPS are strongly interdependent. On the one hand, planning guarantees on the long term the observance of high level temporal and resource capacity constraints and thus sets the goals as well as the resource and temporal constraints for scheduling. On the other hand, scheduling is responsible for unfolding a production plan into executable schedules; i.e., to detailed resource assignments and operation sequences. No scheduling strategy can improve much on an inadequate plan, whereas a bad scheduling strategy that wastes resources may inhibit the fulfillment of a good plan. All this makes PPS extremely complex and hard to solve. At the same time, PPS calls for efficient decision support methods and intuitive, flexible models with fast, reliable solution techniques that scale-up well even to large problem instances. Hence, even if production planning and scheduling problems are solved in a superior-inferior hierarchy, they have to be treated in an integrated manner.

In this paper we review the main goals of PPS and introduce a model of an integrated planner and scheduler system. After having proposed solution principles, we describe an implemented prototype system and the lessons of its experimental use.

2. PROBLEM STATEMENT AND OBJECTIVES

Production planning (PP) is responsible for making the aggregate plan of using production resources (workforce, equipment) and material to meet customer orders. Typically, the plan covers a wide time horizon of several months. Besides giving the date of completing each customer order, production planning determines the capacity and material requirements of production over time. For instance, when materials should arrive, which activities should be outsourced to subcontractors, when to increase or decrease the workforce level. Due to the strong interrelations among such decisions, these questions must be addressed simultaneously. As a consequence, the two strongly coupled, but traditionally separated function of the production planning – Material Requirements/Manufacturing Resource Planning (MRP, MRP II) and Capacity Requirements Planning (CRP) – must be handled in an integrated way.

According to the traditional approach, production planning in make to order production systems is still based on *lead time estimations* and on MRP logic [14]. In this approach, customer orders are segmented with milestones, and the time needed to reach the next milestone is estimated by production lead times. Estimates are based on *past experience* rather than on the actual production load. The MRP system determines the timing of these milestones backward from the due dates of the customer orders. At this stage of the planning process the *actual production capacities* are considered only implicitly, i.e., through the lead times, which again are based on historic data. In the subsequent stage, when the timing of production activities has already been set, production capacities are allocated to the specific customer orders. If in a certain period of time, the in-house capacity is not sufficient to meet demands, decisions are made as whether to extend the capacity of the scarce resource or to involve subcontractors.

Scheduling is responsible for making detailed, executable schedules that achieve the goals set by production plans. Hence, scheduling has to assign finite capacity resources to production operations as well as to determine their order of execution. So as to guarantee that all shop orders can be executed in time and that load on resources never exceeds available capacity, scheduling has to unify the resource and temporal aspects of production at the most detailed level of aggregation. Beyond satisfying various temporal and resource capacity constraints, the solution should approach optimality with respect to some optimization criterion. Close-to-optimal solution of scheduling problems requires expressive and flexible models and efficient, customized solution methods.

In our opinion, the main requirements towards the models and the solution methods of an integrated PPS system are as follows:

- Representation methods at both levels should be able to capture relevant temporal as well as resource capacity constraints of production.
- The results should be optimal or close-to-optimal according to various objectives, and robust to cope with unexpected disorders.
- Production plans should be unfoldable into executable schedules. Hence, planning must also handle precedence relations that ensue from complex product structures (e.g., assemblies) and technological routings.
- However, resource assignment problems with finite capacities and precedence constraints are in general extremely hard to solve. Planning must apply aggregation so that typical instances of planning problems could be solved in a tractable way.
- The solution methods applied at both levels have to be efficient enough to support interactive decision making.
- Both planning and scheduling should use the same master data readily available in *de facto* standard production information systems: product and technology related data (e.g., bills of materials (BOMs), routings), resource calendars, and order data.
- The actual status of production and open orders should be handled on both levels.

3. THE MODEL OF THE INTEGRATED PLANNER AND SCHEDULER

In what follows we propose an integrated PPS system that was designed to meet the above target requirements. The overall framework of the system and its connections to other main modules is presented on Fig. 1 below. Note that the proposed PPS system provides a bridge between *de facto* standard Enterprise Resource Planning (ERP) and Manufacturing Execution (MES) systems. In the overall framework the role of simulation is to validate production schedules and test their sensitivity towards factors that are included neither in the planner nor in the scheduler model.

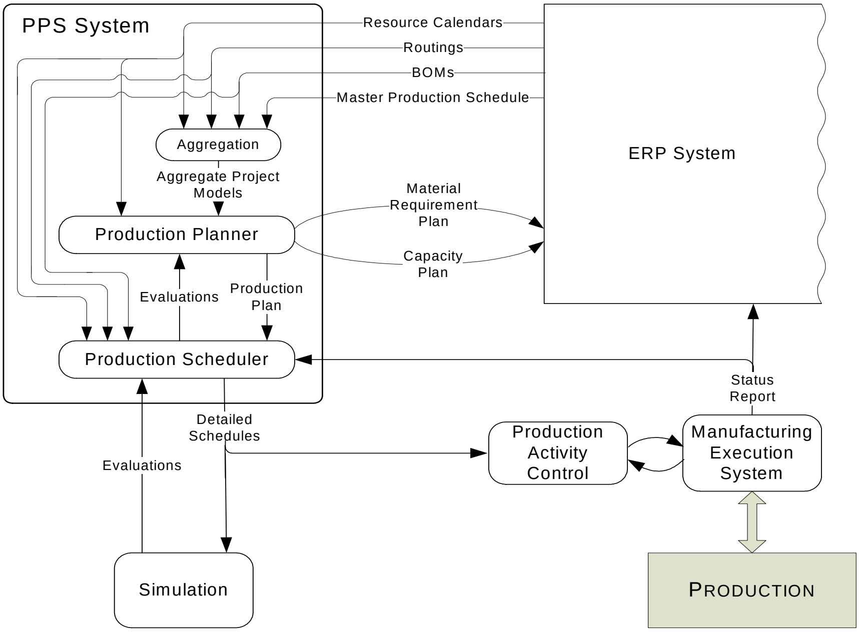


Figure 1: Structure of the PPS framework

3.1. Production planner

The challenge of production planning is the timing of the activities in medium term, over a typically 3-6 months long time horizon with a time unit of one week. The generated plans must comply with the project deadlines, obtain effective utilization of the resources with finite capacity, keep stock levels low, and on the whole, minimize the cost of the production. In our production planner the MRP logic and the production lead times are replaced by a *resource-constrained project scheduling* model. The timing of the competing customer orders is determined with taking into consideration other orders and the production capacity available. In particular, excessive use of certain resources in a time period is possible only if there is no way to avoid this but violating some customer order deadlines. All decisions are made with regard to the actual set of orders and production load. At the same time, important questions − such as the date when certain material should be available, or when additional resources are needed − get answered too.

In the planning problem, *projects* consist of various *activities* needed to complete an order. Usually, some ordering of the activities is to be followed, but many of them may overlap in time, especially in case of large, complex projects. Each activity may call for the use of a number of different resources. The *resources* are typically either machine or human resources (or both, in a coupled way) that will be shared by the activities of different projects. The resources may be distributed, geographically dispersed and may even belong to different organizations.

Each product order is considered a *project*. A project has a *time window* set by the negotiated earliest starting time and deadline. Activities of the same project are linked by *precedence constraints*. An activity may require the execution of a given amount of work on one or more resources. However, the *intensity* of executing an activity may vary over time; the activity can even be pre-empted.

Activities are *aggregates*: they represent groups of manufacturing, assembly, etc. *operations*, some of which are executed simultaneously, some sequentially, and others independently of each other. This leads to a model in which not the durations but only the work amounts of activities are fixed *a priori*. Activities are defined in the course of an aggregation process (see Sect. 3.3) that uses product, production technology and resource availability information. Summing up, the planner works with the following input data (see also Fig. 1):

- specification of customer orders as given in the master production schedule;
- Bill of Materials (BOMs) of products;
- routings (sequence, processing time and resource requirements) of operations; and
- detailed calendar of available resource (machine, workforce) capacities.

The production planner produces the following outputs:

- medium term production plan, which assigns operations to weeks of the planning horizon;
- medium term capacity plan, which specifies the resource requirements of each week on the planning horizon; and
- medium term material requirement plan, which specifies for each week the requirements for raw materials and other components.

3.2. The scheduler

The ultimate goal of scheduling is to unfold the medium term production plan into an *executable* detailed schedule. The scheduler has to determine the order of the operations and the resource allocations with respect to the technological, temporal and capacity constraints. Our short-term scheduler performs finite capacity scheduling with respect to detailed technological and capacity constraints. The scheduling horizon is as long as the time unit of the planner (i.e., one week), while the scheduling time unit is 0.1 hour.

The set of *operations* to be scheduled are determined by disaggregating the activities that fall into a given time unit in the medium-term production plan. If an activity covers several weeks, then its operations are distributed in this period proportional to the activity's intensities. Typically, schedules are generated for the next few weeks only.

Most operations are non-preemptive but breakable, i.e., the workpieces cannot be unmounted from and re-mounted to the machines only after completing the operations. However, the on-going operations can be interrupted, e.g., during the weekends, and continued later on without any extra cost. We model also non-breakable operations (like heat treatment) that must not be broken due to technological reasons.

There are both individual (e.g., machine tools) and group *resources* (homogeneous machine groups, assembly stations, various pools of qualified workforce). Resource availability – that may vary shift-by-shift – is given by the detailed resource calendar. Resource and time requirements, as well as the sequence of operations are described in the *routings*. In the routing, each operation requires a given combination of resources. E.g., a turning operation might require a turning centre and a machinist during the entire length of its processing. In our scheduling model, operations have also specific *processing*, *setup,* and *transportation* times. We assume that transportation and setup are performed before the operation, but while the first needs the workpiece only, setup requires solely the resources of the operation.

The optimization *objective* of the scheduler is to minimize the maximal tardiness with respect to the due dates set by the production plan.

We took the constraint-based approach [1] to model the above scheduling problem. In the constraint model, *variables* are the start times of the operations. The variables are linked by several types of *constraints*. Temporal constraints are the *precedences* between the operations (as given e.g., in the routing, or implied by the BOM of complex products), the *durations* and the *time windows* (earliest start, latest finish times) of the operations. Performing setup and transportation may call for further time constraints, depending on the actual situation. *Resource constraints* prescribe that the resource requirements of the particular operations should be satisfied by limited resource capacities. Therefore, the solution of this problem is an assignment of start times to operations such that all temporal and resource capacity constraints are observed.

Summing up, the scheduler works with medium term production plans, detailed resource calendars, as well as BOMs and routings (see also Fig. 1). In return, it generates detailed predictive production schedules that satisfy all the technological and resource constraints, and approach optimality with respect to the actual optimization criteria.

3.3. Aggregation: the connection between planning and scheduling

Although planning and scheduling models are built by using the same source of master data, the models are different at the two levels. On the one hand the size of the problem, on the other hand the uncertainty of the information related to future events suggest that the production planner should work with an *aggregate model* that covers only the most important temporal and resource constraints of the problem. However, since the production plan must be executable also on the job-shop level, aggregation – the creation of the medium term problem – is a very subtle task [2].

Traditionally aggregation involves the grouping of the operations which belong to the same project and require the same resource into an aggregate activity. This simple approach can be easily understood by human experts, but it can result in very complicated temporal relationships among the activities of an order [5]. These relationships can be expressed by generalized precedence constraints, intensity-curves, overlapping and similar conditions, but they cannot be generated automatically, since the enterprise information systems usually do not contain the necessary data. Consequently, this modeling policy requires the involvement of human experts. Moreover, this approach cannot guarantee executable plans.

In [13] we have proposed a novel method for constructing aggregate models for production planning departing from the detailed technological routings, BOMs and resource calendars. We note that in case of make-to-order manufacturing – when the ordered items are chosen from a catalog – all this information is already available at planning time.

Orders are considered to be independent from each other. An order is modeled by a so called *project tree* – a rooted tree whose vertices with several children denote assembly operations, while those with a single child represent either machining operations or joining a purchased part to the workpiece. The execution of the project over time advances from the leaves towards the root that stands for the finishing operation of the final product. Edges represent strict precedence relations, i.e., the sons of an operation must all be completed before the operation itself could be started.

During aggregation, connected vertices of the project tree are contracted into components that define the activities of the planning model. This *partitioning* of the project tree is called the *aggregate model* of the project. If two operations of the project tree that are connected by a precedence constraint are inserted into the same activity, then this constraint is omitted from the aggregate model. Otherwise, a precedence constraint is posted between the two aggregate activities. Note that the precedence graph of the activities will also form a tree. The resource requirements of an activity are the sums of the processing and setup times of the contained operations per each resource required. Fig. 2 shows two alternative aggregations of the same project tree.

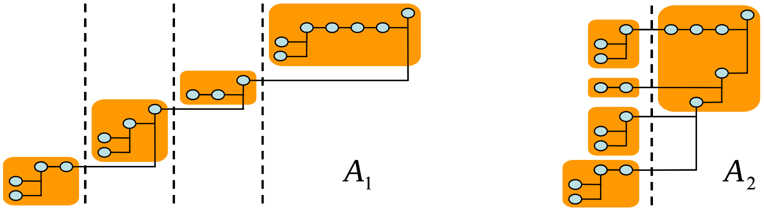


Figure 2: Alternative aggregations of the same project tree into activities

Aggregation has various effects both on planning and scheduling.

- Merging operations of the project tree into larger activities decreases the computational complexity of the planning problem.
- On contrary, too large activities can hardly be unfolded into feasible schedules. Therefore, it is reasonable to set a limit to the size of the aggregate activities. We have proven that the best compromise is setting the size limit of activities to the length of the aggregate time unit (one week, in our case). If an operation with a longer processing time hurts this condition, then this operation constitutes a single activity.
- Though the planning problem is usually considered as a relaxation of the detailed scheduling problem, some extra constraints may be introduced during aggregation. In any case, a precedence constraint in the aggregate model states that the connected activities have to be executed in the given order, *in distinct time units*. Hence, a precedence constraint implies a time unit change between finishing the preceded and starting the preceding activity. Therefore, the lead time of a project using a given activity model cannot be less than the its height. Consider the alternative activity models of the same project at Fig. 2: the two models have different cardinalities (4 vs. 5) and different depths (4 vs. 2). Clearly, *A2* provides a more appropriate aggregation of the same project tree, although it has more activities than *A1*.

Based on the above analysis, the criteria for aggregation are as follows:

- The total resource demand of an activity should not exceed the internal capacity limit per each time unit.
- The height of the aggregate model should be minimal (so that it can contain as many parallel branches as possible).
- The number of activities should be as small as possible (to reduce the problem size).

Since the last two requirements are in conflict, an acceptable trade-off must be found between them.

4. SOLUTION TECHNIQUES AND ALGORITHMS

4.1. Solving the planning problem

When solving the planning problem, our primary objective is the *minimal extra capacity* usage. In this way, the planner attempts to keep the works allotted for a medium-term horizon within the factory. There is also a secondary objective: in order to minimize inventory costs, the level of *work-in-process* (WIP) should be minimal. We note that classical optimization criteria, like *project duration*, *maximum tardiness* or *weighted tardiness* fit also in the proposed framework.

To formalize the planning problem we have used a *resource-constrained project scheduling* model [12], whose detailed analysis has shown, that generally it is NP-hard. However, the analysis resulted also in a linear program re-formulation with cutting planes. The solution method uses them in a custom-tailored efficient *branch-and-cut search* that finds optimal solution [8]. The proposed algorithm is *any-time*: it generates a series of solutions with better and better objective values, thus a feasible solution can be generated quickly and then it can refined to converge towards the optimal one.

4.2. Aggregation

The generation of optimal aggregate project models corresponds to partitioning the project tree into sub-trees that represent activities of the project. The thoughput time of the activities should not exceed the limit of one week, while the height and the cardinality of the partitioning should be as small as possible. In [9] we have suggested polynomial time algorithms for solving such problems.

In order to check whether an activity fits into a week's capacity profile, one has to estimate the throughput time of the set of operations that constitute the activity. However, at the time of activity formation this throughput time can hardly be computed. Firstly, the set of activities competing for the limited resources is not known at this phase. Secondly, determining the minimal throughput time of a single activity is an NP-hard resource-constrained project scheduling problem in itself. Hence, we elaborated various *heuristic functions* for estimating the throughput time of an activity. In production environments where both resource constraints and complex precedence relations should be accounted for, we applied a priority-rule based scheduler that worked with the "greatest rank positional weight*" (GRPW*) rule [4].

4.3. Detailed constraint-based scheduling

As discussed above, we have modeled the scheduling problem as a *constraint optimization problem* [1]. The model uses variables (e.g., start time of operations), possible values (domains) of variables, constraints (resource and temporal) and an optimization objective. When solving this constraint problem, we are looking for those values of all the variables (in their corresponding domains) that satisfy all the constraints and are the best according to the given criteria. Typically, the so-called *maximum-type* objective functions, such as the *makespan* (the maximum of the end times), the maximum tardiness or the peak resource usage can be minimized efficiently by constraint-based techniques.

Solution techniques in constraint programming rely on an effective combination of inference and search. A foundational inference method is *constraint propagation*: it removes inconsistent values from the domains of the variables, i.e., values that provably cannot constitute a part of a solution. Propagation is executed every time the domain of some variable changes. Since the propagation machinery is incomplete, the solution has to be found by a search process. During search, new, artificial constraints are introduced that divide the original problem into separate alternatives. Search decisions and propagation are interwoven so that propagation can reduce the search space as soon as possible. If the constraint model becomes inconsistent in one of the alternatives, then work continues with the other ones, and – in the last resort – the system backtracks.

Constraint-based scheduling applies both the generic propagation mechanisms of constraint programming and domain specific propagators that fit the actual temporal and resource constraints of the scheduling problem. Specifically, temporal constraints between operations can be propagated by versions of the standard, so-called *arc-B-consistency* algorithm [11]. For instance, if a precedence constraint prescribes that operation *A* must be executed before operation *B*, then the earliest start time of *B* should be at least the earliest start time plus the duration of *A*. Once the time window of an operation is reduced, propagation tries to narrow the time windows of all the other ones that are linked to this operation by precedence constraints.

For propagating *resource constraints*, we apply the widely used method of *edge finding* [3]. Given a particular resource and a set of operations requiring this resource, edge finding tries to deduce which operations must be (or cannot be) scheduled first (or last) in this set. The algorithm investigates time windows where the total demand of operations exceeds the capacity of the resource. The conclusions drawn are of two types: new precedence constraints are posted between some operation, and the time windows of some operations are tightened. Note that the application of edge finding may prompt the further call of temporal propagators and *vice versa*.

Even though constraint propagation can help prune the search space significantly, scheduling of discrete resources is still a very hard problem to solve to optimality. Hence, we have embedded constraint propagation into a search process that produces a sequence of better and better solutions that converge to the optimal schedule. The solution method is – like that of the medium term planner – *any-time*, thus it can be used interactively. Some further methods that we applied to increase the efficiency of these solution techniques are described in detail in [10].

5. IMPLEMENTATION AND INDUSTRIAL APPLICATION

The above PPS framework has been developed in the course of the "Digital Factory, Production Networks" NKFP project. During this work, a prototype PPS systems – the so called Proterv-II – has also been implemented that supports production and capacity planning on medium term and detailed job-shop schedule on short term. Proterv-II has graphical user interface that facilitates its use as a *decision support system* (DSS) at both levels of the decision hierarchy. To implement the suggested algorithms, we have used professional constraint solver and optimization software [6].

Experiments with the prototype system have been carried out on real-world industrial data. Typical projects consisted of 20 to 500 discrete manufacturing operations, with processing times in the range of 0.5 to 120 hours. Operations required both machine and human resources. The project trees were generated from standard BOM and routing databases. The aggregate project models were generated from these trees consisted of 1 to 10 activities. The resource pool contained ca. 100 individual and 50 group resources. The horizon of the planning problem was set to 15-30 weeks. Under the above initial conditions, our approach was capable to generate optimal production plans even for several hundreds orders. The first feasible solution of the studied planning and scheduling problems could be created within a few seconds, and if one looked for the optimal plan, larger runtime had to be set.

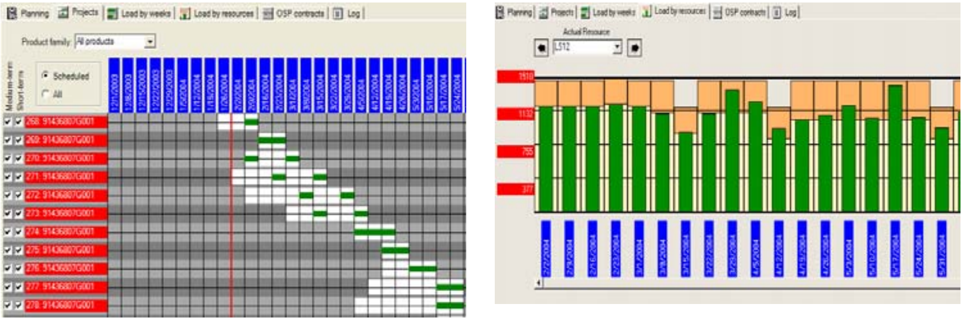
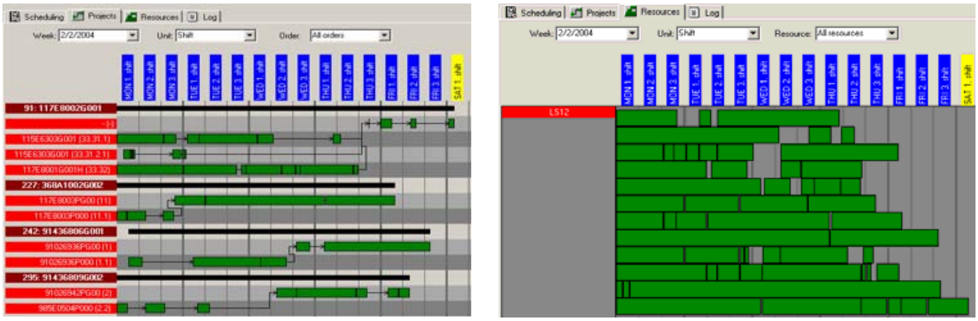


Figure 3: Medium term plan and load of a resource

Fig. 3 presents a fragment of a production and a capacity plan. In the production plan the white areas show the allowed time windows of the projects and dark fields indicate weeks when one or more activities of the projects have to be performed. Since the minimal WIP level was a criterion, the projects start as close to their due dates as possible without violating them. The breaks in the production are caused by resource shortages. The capacity plan shows the load of a certain resource through the planning horizon. It looks really to be an overloaded factory: the internal capacities are completely exploited and in several weeks the use of extra capacities is also required.

In Fig. 4 a segment of a short time schedule is presented. The schedule contains the operations of the activities – connected parts of the project trees – distributed in the week, while the resource view shows which operations have to be performed on a certain resource or resource group.


Week: 2/2/2004 Unit: Shift Order: All orders

91: 117E8002G001
-(-)
115E6303G001 (33.31.1)
115E6303G001 (33.31.2.1)
117E8001G001H (33.32)
227: 368A1002G002
117E8003PG00 (11)
117E8003P000 (11.1)
242: 914368066001
91026936PG00 (1)
91026936P000 (1.1)
295: 91436809G002
91026942PG00 (2)
985E0504P000 (2.2)

Week: 2/2/2004 Unit: Shift Resource: All resources

LS12

Figure 4: Short term schedule and load of a machine group

During the execution of the produced schedules various disorders (e.g., machine breakdowns or distortions come from model uncertainties) may occur. These situations have been analyzed and evaluated by discrete event simulations [7]. The simulation experiments have shown that in many cases the medium term plans are robust enough to remain feasible despite numerous unexpected events. The operations which could not be executed at the proper week could be included in the schedule of the next week without violating the deadlines.

Our model can also be enriched with a Production Activity Control (PAC) module (see Fig. 1), whose purpose is to support the execution of schedules under dynamic, ever-changing conditions at the shop floor. In a realistic production environment a rigid schedule with fixed operation starting times can hardly be executed. By removing the fixed start times of operations and keeping only the sequence of the operations, queues can be formed in front of each resource. Operations can be picked by the application of a dispatching rule that keeps the queues all the time consistent with the original precedence constraints. As a reaction to smaller changes in the environment, the PAC module should perform synchronization – a re-scheduling based on the operation queues with respect to the newly emerged constraints – by the application of simple and fast modifications that cause only minimal perturbation to the original schedule. However, shop foremen should have the final word in deciding on schedule execution: they have the responsibility to postpone or remove some operations from the queues and to ask for a global rescheduling in case of major disorders.

6. CONCLUSIONS

In this paper we have presented an overview of the main concepts and solution methods of a hierarchical production planner and scheduler system. Novel feature of our approach is that it takes a project-oriented approach for solving the planning problem. Hence, decisions on the time of making the customer orders are combined with decisions concerning the load on resources. Further on, these decisions are made with regard to the actual demand and available production capacities. This approach results in better due-date observance and executable production plans.

We have also analyzed the role of aggregation that links models of production planning and scheduling. The proposed aggregation method enables PPS to work on common product, resource and production technology data on both levels of the decision hierarchy. Our experiences confirm also that proper aggregation is a major prerequisite for generating production plans that can really be refined to executable schedules.

ACKNOWLEDGEMENTS

This work has been supported by the NKFP grants No. 2/040/2001, 2/010/2004 and the OTKA grant No. T046509. The authors would like to thank Ferenc Erdélyi, Tamás Kis and László Monostori for their help and support.

REFERENCES

[1] BAPTISTE, PH., LE PAPE, C., NUIJTEN, W.: *Constraint-Based Scheduling*. Kluwer Academic Publishers, 2001.

[2] BITRAN, G.R., TIRUPATI, D.: *Hierarchical Production Planning*. In: Graves, S.C., Rinnooy Kan A.H.G., Zipkin, P.H. (eds), Logistics of Production and Inventory, North Holland, 1993, pp. 523-568.

[3] CARLIER, J., PINSON, E.: *A practical use of Jackson's pre-emptive schedule for solving the job-shop problem*. Annals of Operations Research, 26, 1990, pp. 269-287.

[4] DEMEULEMEESTER E.L., HERROELEN, W.S.: *Project Scheduling: A Research Handbook*. Kluwer Academic Publishers, 2002.

[5] HACKMAN, S.T., LEACHMAN, R.C.: *An Aggregate Model of Project-Oriented Production*. IEEE Transactions on Systems, Man, and Cybernetics, 19, 2, 1989, pp. 220-231.

[6] *Ilog Scheduler 5.1 Users Manual*. 2001.

[7] KÁDÁR B., PFEIFFER A., MONOSTORI L.: *Discrete Event Simulation for Supporting Production Planning and Scheduling Decisions in Digital Factories*. Proceedings of the 37th CIRP International Seminar on Manufacturing Systems, 2004, pp. 441-478.

[8] KIS T.: *A Branch-and-Cut Algorithm for Scheduling Projects with Variable-Intensity Activities*. Mathematical Programming, 2005 february.

[9] KOVÁCS A., KIS T.: *Partitioning of Trees for Minimizing Height and Cardinality*. Information Processing Letters, 89, 4, 2004, pp. 181-185.

[10] KOVÁCS A., VÁNCZA J.: *Completable partial solutions in constraint programming and constraint-based scheduling*. In Proc. of the 10th International Conference on Principles and Practice of Constraint Programming (Springer LNCS 3258), 2004, pp. 332-346.

[11] LHOMME, O.: *Consistency techniques for numeric CSPs*. In Proc. of IJCAI'93 - the 13th International Joint Conference on Artificial Intelligence, 1993, pp. 232-238.

[12] MÁRKUS A., VÁNCZA J., KIS T., KOVÁCS A.: *Project Scheduling Approach to Production Planning*. CIRP Annals - Manufacturing Technology, 52, 1, 2003, pp. 359-362.

[13] VÁNCZA J., KIS T., KOVÁCS A.: *Aggregation – The Key to Integrating Production Planning and Scheduling*. CIRP Annals - Manufacturing Technology, 53, 1, 2004, pp. 377-380.

[14] VOLLMANN, T. E., BERRY W.L., WHYBARK D.C.: *Manufacturing Planning and Control Systems*. McGraw-Hill, 1997.