Conceptual Models for Combined Planning and Scheduling

Roman Barták[*]

Charles University, Faculty of Mathematics and Physics
Department of Theoretical Computer Science
Malostranské námestí 2/25, 118 00 Praha 1, Czech Republic
e-mail: bartak@kti.mff.cuni.cz

Abstract

Planning and scheduling attracts an unceasing attention of computer science community. Several research areas like Artificial Intelligence, Operations Research and Constraint Programming joined their power to tackle the problems brought by real industrial life. Among them Constraint Programming plays the integrating role because it provides nice declarative capabilities for modelling and, at the same time, it can exploit directly the successful methods developed in AI and OR.

In this paper we analyse the problems behind industrial planning and scheduling. In particular we give a survey of possible conceptual models for scheduling problems with some planning features. We compare their advantages and drawbacks and we explain the industrial background. These models were studied within the VisOpt project whose task is to develop a generic scheduling engine for complex production environments.

Key words: scheduling, planning, modelling, constraint satisfaction, optimisation

1 Introduction

Planning and scheduling attract high attention among researches from the areas like Artificial Intelligence (AI), Operations Research (OR) and, recently, Constraint Programming (CP). Roughly speaking, planning deals with finding plans to achieve some goal, i.e., finding a sequence of actions that will transfer the initial world into one in which the goal description is true. Industrial planning usually means finding a plan of production for a longer period of time. From this point of view, scheduling can be seen as a more detailed planning for shorter period of time. More precisely, scheduling deals with the exact allocation of the resources to the activities over time respecting the precedence, duration, capacity and incompatibility constraints.

---

[*] Supported by the Grant Agency of the Czech Republic under the contract no 201/99/D057.

26 September 1999

Operations Research has a long tradition in studying the scheduling problems and many successful methods to deal with the problem were developed. Recently, Constraint Programming attracts high interest among scheduling community because of its potential for declarative description of problems with various real-life constraints.

Constraint programming [5] is based on the idea of describing the problem declaratively by means of constraints, logical relations among several unknowns (or variables). In the second stage, the solution, i.e., an assignment of a value to each unknown from respective domain, is being found in such a way that all the constraints are satisfied. It is possible to state constraints over various domains, however, currently probably more than 95% of all constraint applications deal with finite domains [18]. And among them, the scheduling problems are the most successful application area [14,19].

Constraint programming represents one of the closest approaches computer science has yet made to the Holy Grail of programming: the user states the problem, the computer solves it [E. Freuder, Constraints, April 1997]. However, relying on this statement is also the biggest danger of real-life projects based on constraints because the generic constraint satisfaction and optimisation algorithms are still not capable to tackle efficiently large-scale industrial problems brought by real-life. At least without additional help. Thus, the constraint modelling, i.e., a description of the problem by means of constraints, is very important part of all projects. Also, Operations Research and Artificial Intelligence are bringing their checked methods to improve the efficiency of constraint systems using global constraints [16] and more advanced search techniques.

In this paper we give a survey of basic techniques, concepts and mechanisms developed within the project of generic scheduling engine with planning capabilities. The task was to prepare a generic model capable to capture various planning and scheduling problems in the complex production environments. We summarise the results of the first stage of the project where three basic conceptual models to represent the problem were studied and the core constraints describing the problem area were introduced. At this stage we concentrated on the expressive power of the models primarily, we give a description of industrial background behind the models, and we only touched slightly the efficiency issues. Next stage of the project will cover the definition of additional, mostly redundant constraints for value propagation as well as a special labelling procedure. At this stage, the efficiency of the system becomes the main issue.

2 Problem Area

In the VisOpt scheduling project [4] we deal with the complex production areas like plastic, petrochemical, chemical or pharmaceutical industries. The task is to develop a generic scheduling engine with planning capabilities for the complex production environments. This engine should be customisable easily for particular production environment via the description of resources.

The problem domain can be described as a complex heterogeneous environment with *several resources* interfering with each other. Currently we are working with producers, movers and stores, later other resources like workers and tools will be added. Some resources can handle several tasks at a time (during *batch production*) and the task can be scheduled to multiple alternative resources. The processing time is variable and the processing in a resource is restricted by the *working time* specified for the resource.

Alternative processing formulas, alternative production routes and *alternative raw materials* are other typical features of above mentioned industry areas. Also, there is a possibility to produce the *by-products* and the *co-products* that should be scheduled for later consumption as raw material (*re-cycling*) or can be sold as an alternative to the ordered product. Because all products can be stored in identical warehouses, the capacity and compatibility constraints play important role. There is also possibility of *cycling*, i.e., processing the item several times for example to clean up the store or to change features of the item. We should consider the transport and the waiting times as well as we should follow the *set-up* and *transition patterns*.

Typically, the production is not driven by the custom orders only but it is possible to schedule the production for store, i.e., not everything that is produced was ordered. Therefore the system should manage some planning capabilities as well. The task is to generate the most profitable schedule.

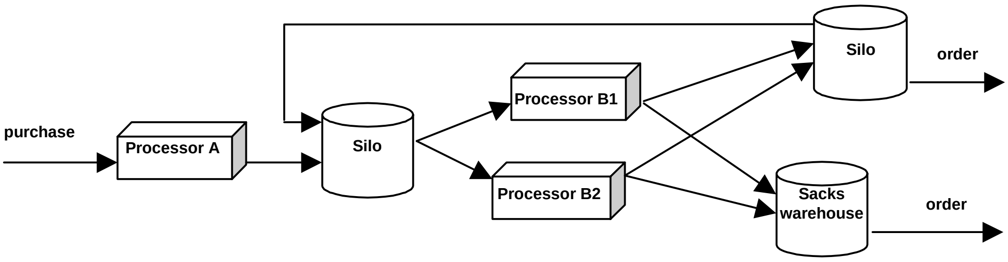

Fig. 1. The example of complex production environment

The solved problem is close to the group of resource constrained project scheduling problems (RCPSP). The RCPSP [9,12] is a generalisation of job shop scheduling [3] in which tasks can use multiple resources, and resources can have capacity greater than one (more tasks can be processed together). Nevertheless, the RCPSP is still defined by a set of tasks and by a set of resources, and both sets are known in advance. Unfortunately, this is not our case because when the alternative processing routes, the by-products and the co-product, the cycling and the re-cycling and the production for store are assumed then the set of all tasks/activities is not known beforehand. Also, it is not possible to use a foregoing planning stage to generate the set of tasks before the scheduling because of the close relations between the resources that require detail scheduling. Consequently, the system must be a mixture of both scheduling and planning components.

3 Conceptual Models

In general, it is a good design principle to create a declarative and thus transparent model of the problem. All entities are created initially with constraints that define their nature and the relationships between them. The search code is kept separate from this declarative model. This is exactly what we have done by designing the conceptual model that describes the representation of the resources and their limits.

In the project we assumed three conceptual (declarative) models of the scheduling problem: the time-line model uses a method of time slices to discretise the time, while the other two models, namely order-centric and resource-centric models, use traditional notion of activity.

3.1 Time-Line Model

The first conceptual model that we studied in the project is a time-line model. The time-line model is a general method of describing dynamic processes using discrete time intervals. We first divide the time line into a sequence of time slices with identical duration and at each time point (the point between two slices) we describe the situation of each resource using several variables, for example, what is currently produced or stored. It is assumed that the behaviour of the resource is homogeneous between two time points (within the time slice), i.e., the key events like changing activity occur only at the edge of two consecutive time slices.

The duration of time slices must be defined according to the duration of the activities that can be processed by the resources. Because of the slice homogeneity, it is required that the activity change is between two consecutive time slices only. Consequently, each activity takes one or more time slices but no fragment of single slice. We can compute the duration of the time slice for given resource as a common divisor of the duration of all activities that can be processed by the resource.

Also we need to synchronise different resources so we prefer the time slices to have the same duration in all resources. This refines the slices further because the duration of time slice should be equal to a common divisor of the duration of all activities in all resources. On the other hand we prefer a smaller number of slices (i.e., a longer duration) because it implies a smaller number of variables to assign a value, i.e., less work to do. Together, the *duration of the slice* is computed as the greatest common divisor of the duration of all activities in all the resources.

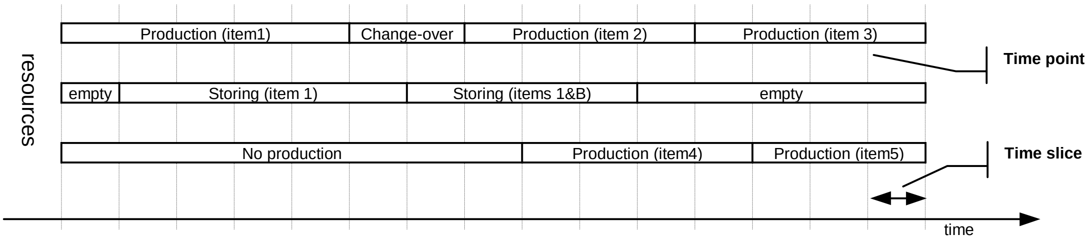

Fig. 2. The time-line model

Now, when the time slices are defined we can describe the situation of each resource on the edge between two consecutive time slices. We call this edge a time point. The situation is described simply by using a set of variables. For example in case of the store there is a variable for each item that can be stored and this variable specifies the stored quantity. Other variables can specify the state of the resource etc. Then, we can introduce constraints between these variables that describe the resource features, like the compatibility and the capacity constraints.

***The capacity constraint*:**

$\forall T \quad \sum_{items} Quantity(R,T,Item) \leq Capacity(R)$, where T is a time point and R is a resource.

The sum of quantities of all procesed items does not exceed the capacity of the resource.

***The compatibility constraint*:**

$$\forall T \ \ (Quantity(R,T,Item1) = 0 \vee Quantity(R,T,Item2) = 0)$$

If Item1 is incompatible with Item2 then they cannot be stored or processed together.

In most cases, the current situation of the resource is influenced by the previous situation of the resource as well as by other resources. Again, there is no problem to define the constraints between the variables from different time points and even from different resources. For example, in case of store, the current quantity of the item is influenced by the previous quantity, by new supplied quantity (the item is supplied to the store) and by the consumed quantity (the item is removed from the store).

The supplier/consumer constraint:

$$
\begin{aligned}
&Quantity(R,T,Item)\\
&\quad = Quantity(R,T-1,Item) + \sum_{S} Quantity(S,T-X_{S,R},Item) - \sum_{C} Quantity(C,T+X_{R,C},Item)
\end{aligned}
$$

The quantity of Item in the resource R in the time T is computed using the quantity in time T-1 plus the sum of supplied quantities minus the sum of consumed quantities. $X_{A,B}$ is a transport time between the resources A and B expressed in multiples of slice duration.

The above supplier/consumer constraint also shows why we need to synchronise the slice duration among the resources. This is because we need a common “clock frequency” for the description of the relation between situations of different resources. Then, we can measure the time by ticks, i.e., by the number of slices, instead of using real-time unit (seconds etc.).

The model of scheduling problem can be represented as a matrix of variables where one axis corresponds to the parameters describing the resources and the second axis corresponds to the time points (the number of time points specifies the scheduled duration).

resource	parameter	variables							
polymer	state								
	…								
extruder	state								
	…								
…									
store	Item 1 quantity								
	Item 2 quantity								
	…								
	time points	1	2	…					N

Fig. 3. The matrix representation of the time-line model

In the time-line model, there is no problem to represent the initial situation as well as to capture any required future situation. We can simply set the values of variables in the time point 0 to describe the initial situation and we can restrict the domains of the variables in future time points to capture the desired future situations. Naturally, such domain restriction is valuable because it can navigate the scheduling and decrease the size of the search space. Also, we can use the same mechanism to implement heuristics derived from user’s experience like the specification of the minimal stored quantities of items etc.

The time-line model is very general and because of its nature it is capable to capture all planning and scheduling situations. The transparency of the model is its biggest advantage. Another advantage is that all the variables and the constraints among variables are defined before the scheduling starts. Consequently, we can use arbitrary constraint satisfaction and optimisation technique including local search methods to solve the problem.

Unfortunately, the model has one big disadvantage that prevents its usage in real large-scale problems. This drawback is hidden in the definition of the time slice duration. In most cases we can expect that this duration will be very short, even if the activities take long time. For example, if there are activities with the duration 25 and 26 seconds respectively, we still have to define the time slice to have the duration 1 second because we don't know the order of activities beforehand. As a consequence, there are a large number of time points that implies huge number of variables to label. Therefore, we can expect not very good efficiency from the model if applied to the real-life scheduling problems. However, we believe that the model can be applied successfully in cases when:

- the description of resources is not very complicated, i.e., we use a small number of variables to describe the situation,
- the ratio between the time slice duration and the scheduled duration is higher, i.e., either the scheduled duration is short or the duration of the time slice is long.

Because of efficiency issues, we turned our attention to a more classical representation of the scheduling problems using activities.

3.2 Order-centric Model

Order-centric model[1] is a traditional model for the job-shop scheduling [3]. It is based on the idea of defining the chain of activities necessary to produce the ordered item or, generally, to satisfy the order. The goal is to schedule such production chains for all the orders respecting the resource limits. In Fig. 4 we show an example of the production chain. You may notice that by the notion of the production chain we mean not only a linear sequence of activities but also, for example, a tree of activities with the root corresponding to the final product or the order.

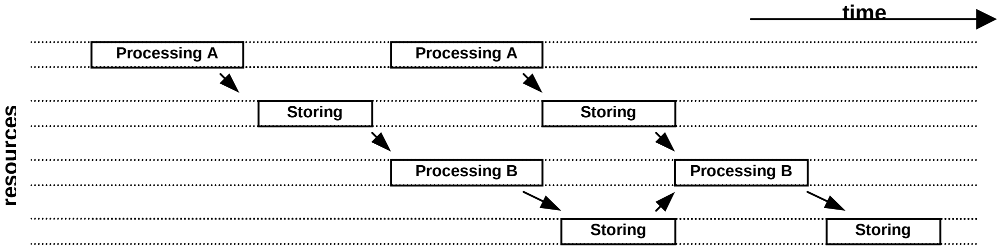

Fig. 4. The production chain in order-centric model

In the order-centric model, we describe activities using a set of variables. Typically, the start and end times (or the duration) are assigned to each activity as well as the resource where the activity will be performed. The value of these variables is being found during scheduling. Other parameters, like a processed quantity or a specification of the activity are usually constant because we know what quantity of the item is ordered. But if a variable formula is used then the quantity can be variable as well.

In general, there are two groups of constraints used in the order-centric model to restrict the combinations of the values in the variables. The first group consists of the constraints

---

[1] Sometimes, the model is called task-based model [7] but we prefer the notion of order-centric model as we assign a production chain to the order.

between the activities within single production chain. We call these constraints *supplier/consumer dependencies* because they describe the movement of items between the resources (supplying and consuming). Depending on the parameters describing the activities, there may be the timing and the quantity constraints between the activities. The timing constraint specifies the order of activities in the production chain, i.e., the precedence of former activity before later activity in the chain. The quantity constraint is used if variable quantities of the item can be produced. This constraint binds the produced and the consumed quantities in consecutive activities.

The timing constraint:

$$end(Activity\ 1) + transport\_time(resource(Activity1), resource(Activity\ 2)) = start(Activity\ 2)$$

if Activity 1 foregoes Activity 2 in the production chain.

The quantity constraint:

$$quantity(Activity\ 1, Item) = quantity(Activity\ 2, Item)$$

for each item produced in Activity 1 and consumed in Activity 2.

The second group of constraints describes features and limitations of resources; therefore we call them *resource constraints*. These constraints are evoked if the activity is assigned to the resource only. There may be a constraint binding the resource with the activity duration (it is used if the processing time depends on the resource). Also there are usually the capacity (cumulative) and the compatibility constraints restricting the set of activities that can be processed together in single resource. We can use these constraints to model disjunctive scheduling as well (if we set a different specification to each activity then only one action is allowed to be processed at time)

The compatibility constraint:

$$specification(Activity\ 1) \neq specification(Activity\ 2)\ \&\ resource(Activity\ 1) = resource(Activity\ 2)$$

$$\Rightarrow$$

$$end(Activity\ 1) \leq start(Activity\ 2) \vee start(Activity\ 1) \geq end(Activity\ 2)$$

If Activity 1 and Activity 2 are processed by the same resource and they are incompatible then they cannot be processed together.

The capacity constraint:

$$\forall T\ \forall R \qquad \sum_{\substack{Activity \\ start(Activity) \leq T \leq end(Activity)\ \&\ resource(Activity) = R}} consumed(Activity, R) \leq Capacity(R)$$

At each time the capacity of the resource is not exceeded.

The order-centric model can be represented as a set of production chains where each production chain is represented as a list of activities. Note that even if we use more complex structure of the production chain, like a tree, it is still possible to sort activities in some way to get the list. For example, we can sort the activities using the earliest start time and, in case of clash, use the latest start time or random order.

The production chains may have different length and the activities may be specified using different sets of variables. This is not a problem if we use a list data structure connecting different elements (activities). In Fig. 5 we normalise both the length (to the longest chain) and the sets of parameters (to the superset of all parameters) to get a matrix view of the model. In such case, some variables are void during scheduling (we may set them to zero).

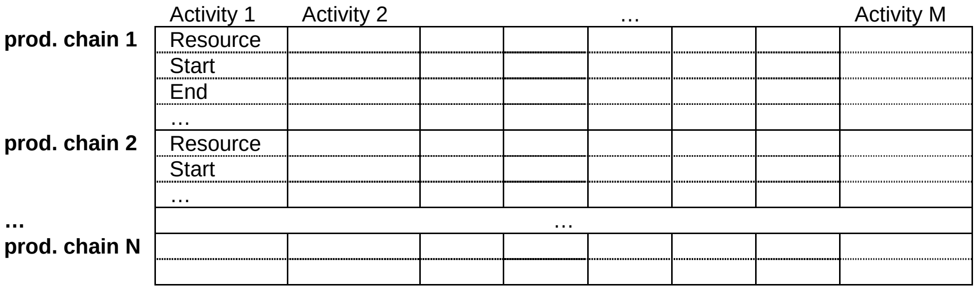

Fig. 5. The matrix representation of the order-centric model

We identified several problems of the order-centric model in the context of VisOpt scheduling project.

The first problem is using the *alternative processing routes* that are typical for chemical industry. This means existence of several ways how to produce the item (satisfy the order), i.e., instead of single production chain we have a set of more or less different production chains per order. The typical example is inserting re-heat activity if the whole chain of activities takes too long [15]. It is possible to extend the order-centric model to work with the alternative production chains by means of virtual activities. Similarly to Fig. 5 we may define the virtual production chain with the length equal to the length of the longest alternative production chain. The virtual activity in this chain represents several activities from the alternative production chains. To model the virtual activity we can add a new variable to the activity whose value indicates the chosen activity. Of course, we need to normalise the other parameters of the activities in the production chain like in Fig. 5 and to adjust the constraints.

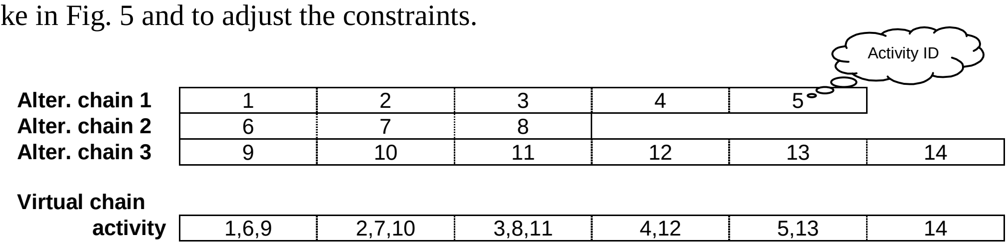

Fig. 6. The representation of the alternative production chains

Now, if we select one activity during the scheduling then this information is propagated to other virtual activities in the chain and the corresponding constraints evoke. For example, if we assign 6 to the activity parameter in the first virtual activity then we can assign 7 and 8 to this parameter in the second and third virtual activity respectively and make remaining virtual activities empty. Nevertheless, this solution complicates the constraints significantly if more alternatives are available. Also, it does not solve the problem of satisfying single order using several (parallel) production chains, e.g., instead of producing 1 ton, we produce 500 kg two times etc.

The second problem of the order-centric model is modelling the *set-up times* [15] and other more complicated resource constraints. The set-up time specifies the necessary duration/gap between consecutive activities processed in single resource and, consequently, it depends on the order of activities chosen during the scheduling. The set-up time can be modelled by introducing special set-up activities with specified duration dependent on the previous activity in the schedule.

The next problem is the processing of the *by-products* and the *co-products*, i.e., the production of non-ordered items that can be re-used in the future production. Now, the situation is more complicated because it requires two or more production chains to interact in a more advanced way, i.e., if a by-product is produced in one production chain then another production chain can use this by-product as a raw material. We propose to model handling of the by-products passively using alternative production chains. If there is a production chain that can use the by-product as an alternative to the raw material then this production chain can be chosen only if the by-product is in a store already, i.e., if we scheduled production of the by-product already in other production chain. Notice that in the above case two production-chains share single storing activity. It means that during scheduling of one chain (the by-product supplier) we set the start time of storing and during scheduling of the second chain (the by-product consumer) we set the end time of storing.

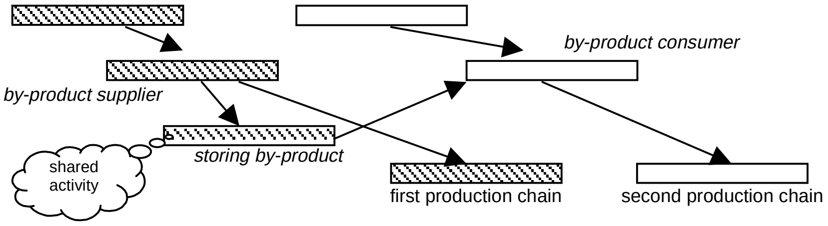

Fig. 7. Sharing activities

Finally, there is a problem with the *production for store*, i.e., non-ordered production. Unfortunately, this problem cannot be solved in pure order-centric model fully because there are no production chains for non-ordered items. One of the possibilities, how to model the non-ordered production, is using foregoing planning stage that generates virtual orders for the items not ordered by the customers. However, there is a problem with the interaction between the planner and the scheduler because the planner may ask to schedule the production for a virtual order that could be in conflict with other orders. Therefore a closer co-operation between the planner and the scheduler is desirable.

The order-centric model is a perfect model for order-driven production. However, as soon as more resource constraints are required, this model becomes more complicated. And if we need to join scheduling with planning stages then this model is of little help. Therefore we turned our attention to the resource-centric model.

3.3 Resource-centric Model

The resource-centric model is similar to the order-centric model in the way of using the activities. However now we are working with the list of activities per single resource and the chain of activities per order is handled implicitly by means of dependencies between the resources.

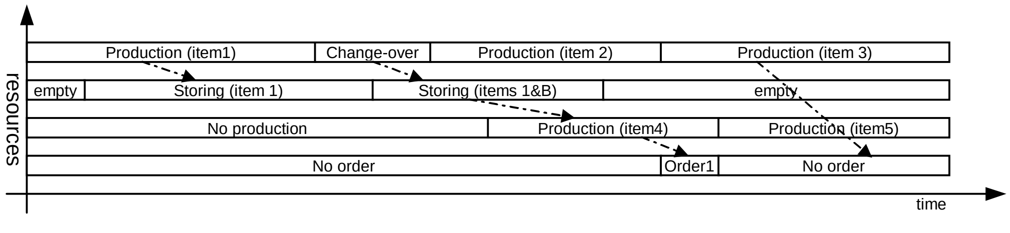

Fig. 8. The resource-centric model

The resource-centric model is more appropriate for the description of a factory than the order-centric model. This is because we concentrate on the specification of resources and this specification is independent of the actual set of orders. This is similar to the time-line model, where we also describe the resources primarily.

Each resource in the model is described using the set of activity types[2] that can be processed by the resource and using the resource constraints. The parameters of the activity (the activity type) are the same as in the order-centric model with two exceptions:

- we do not need the parameter identifying the resource for the activity because each activity is assigned to the resource implicitly,
- we need the parameters identifying the supplying and the consuming activities.

The resource-centric model can be represented as a set of resources where a list of activities is assigned to each resource. The order of the activities in the list corresponds to the order of the activities in the schedule but we know neither the start and end times nor the activity type. Assigning values to these unknowns is the core of scheduling.

Again, we can use the virtual activity to model the choice of activities to be processed. The question is how many activities can be processed by the resource within the scheduled duration. In the real life applications there is usually a minimal duration of the activity defined. So we can compute the upper bound of the number of activities processed by the resource using the formula:

$$\mathit{upper\_bound} = \mathit{scheduled\_duration} \div \mathit{min\_activity\_duration}.$$

Note that if the activities can be processed in parallel then we should multiply the upper bound by the maximum number of parallel activities. Now we can generate the list of activities for each resource and the length of the list equals to the upper bound. In Fig. 9 we normalise the number of activities per resource to the greatest upper bound among resources.

	Activity 1	Activity 2			…			Activity M
resource 1	Activity type							
	Start							
	End							
	…							
resource 2	Activity type							
	Start							
	…							
…					…			
resource N								

Fig. 9. The matrix representation of the resource-centric model

Naturally, it is easier to express the *resource constraints* in this model because we know which activities belong to the resource. There is no problem with the alternative production routes because all possible activities are specified within the resource so the alternative production chains will be composed from these activities implicitly during the scheduling (by setting the suppliers and the consumers of the activity). The set-up times can be modelled using the set-up activities and the by-products are handled in the same way as other products. In fact, we do not distinguish between a by-product and a regular product because everything that is produced by some resource must be consumed by another resource. Visibly, we can also schedule the production without the orders but if any order is present then it is used like a consumer of the ordered products. We must schedule the resources in such a way that the ordered products are available at specified times.

---

[2] The relation between activity type and activity is the same as the relation between class and object in OOP.

One question remains to answer only: how to model the *supplier/consumer dependencies*? Because we know the structure of the factory, we know how the resources are connected. Consequently, for each item we can identify the group of supplying activities and the group of consuming activities from respective resources. Now we may add a list of variables to each supplying activity specifying the quantities transferred to respective consuming activity and vice versa. The parameter *supplied(X,Y)* in the resource X represents the supplied quantity from the resource X to the resource Y and the parameter *consumed(X,Z)* in the resource X represents the quantity consumed by the resource X from the resource Z. Then the supplier/consumer dependency consists of three constraints:

- the sum of supplied quantities is equal to the produced quantity and the sum of consumed quantities equals to the overall consumed quantity,

$$\sum_{X \in \text{consumer\_of(A)}} sup\ plied(A,X) = produced(A)$$

$$\sum_{X \in \text{supplier\_of(B)}} consumed(B,X) = consumed(B)$$

- the quantity in A supplied to B equals to the quantity in B consumed from A,

$$supplied(A,B) = consumed(B,A)$$

- if the interchanged quantity is greater than 0 then the time difference between the end of the supplying activity and the start of the consuming activity is equal to the transport time (because of previous constraint we can use both the *supplied* and *consumed* parameter).

$$supplied(A,B)>0 \Rightarrow end(A) + transport\_time(A,B) = start(B).$$

The resource-centric model is general similarly to the time-line model because it is based on the description of the resources rather than on the description of the orders. It is also more efficient than the time-line model because we can use different scheduling resolution[3] in different resources. Consequently, a smaller number of variables is introduced to the system. Finally, the resource-centric model can process the orders but it does not rely on the orders completely. So, it can be used to manage both the planning and scheduling tasks.

4 Final Remarks

In most current APS (Advanced Planning and Scheduling) systems the planning and scheduling components are implemented separately in different modules. The planning module is responsible for preparing the plans, i.e. the sequences of the activities to satisfy the orders. The scheduling module schedules these activities, i.e. it assigns the activities to the resources and it determines the exact start and end times of the activities as well as other parameters of the activities. If the scheduler finds that it is not possible to schedule all the activities then it backtracks to the planner to find another plan. This decreases the overall performance of the system, if the planner produces too tight (hard to schedule) plans. On the other hand, if the planner produces too easy plans then the profit of the schedule is smaller than it could be because the resources are not utilised fully.

---

[3] The resolution of scheduling is defined by the minimal duration among activities of the resources.

Because of above reasons we propose to combine both the planning and scheduling components into a single conceptual model. This model performs the scheduling task primarily but it has some planning capabilities as well. Namely, it is possible to choose the alternative production routes and to schedule non-ordered production.

In the paper we give a survey of three conceptual models for planning and scheduling problems. We intend these models for complex production environments and therefore we have to include some features outside the area of traditional resource constrained project scheduling problems like the production of non-ordered products etc.

The *order-centric model* is probably the most widely used model within the scheduling community. This model is suitable for scheduling in the environments where the production chains are know in advance and there is a small number of alternatives or the alternatives can be selected before the scheduling starts. We show that this model is less appropriate for the complex production environments with complicated resource constraints, e.g., transition patterns. We sketch some techniques how to solve the problems in such environments, nevertheless, these techniques complicate the model significantly and they make it less transparent. Therefore we propose the *resource-centric model* to capture the complex production environments with alternative processing formulas, complex resource constraints and non-ordered production. Finally, the *time line model* is presented here which is the most general one of the proposed models. It is capable to solve the planning and scheduling problems in the complex production environments and to capture many features. The huge number of variables is its main disadvantage. We still believe that this model can be applicable to problems where the description of the resources is not very complicated. It can also be used as a complementary model to the resource-centric model for the description of parallel processing, e.g., to model the store.

We have described the static representations for all three models; i.e., the variables and the constraints are introduced before the scheduling starts. Such representation has the advantage that arbitrary constraint satisfaction method can be used including local search methods. On the other side, the static representation complicates the constraints in case of alternatives because we need to normalise, to use the virtual activities etc. One of the ways to solve this problem is using a dynamic representation where the time points and the activities respectively are generated during the scheduling. This is a more natural way of combining the planning and scheduling modules because as soon as the planner generates an activity, the scheduler tries to schedule this activity so possible clashes are discovered earlier. We think that using a dynamic representation is a promising way to combine the planning and scheduling concepts into a single framework.

Finally, it should be noted that we concentrate on the expressiveness of the proposed models in the paper and further research concerning the implementation details like the propagation and labelling techniques is necessary to get a complete model for the real-life project.

Acknowledgements

The author’s work has been supported by the Grant Agency of the Czech Republic under the contract number 201/99/D057 and by InSol Ltd. I would like to thank Yossi Rissin and the team of InSol for introducing me to the problem and for interesting and encouraging discussions concerning the real-life problems of industrial planning and scheduling.

References

[1] Baptiste, P., Le Pape, C.: A Theoretical and Experimental Comparison of Constraint Propagation Techniques for Disjunctive Scheduling, in: *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Canada, 1995

[2] Baptiste, P., Le Pape, C.: Disjunctive Constraints for Manufacturing Scheduling: Principles and Extensions, in: *Proceedings of the Third International Conference on Computer Integrated Manufacturing*, Singapore, 1995

[3] Baptiste, P., Le Pape, C., Nuijten, W.: Constraint Based Optimisation and Approximation for Job-Shop Scheduling, in *Proceedings of the AAAI-SIGMAN Workshop on Intelligent Manufacturing Systems*, IJCAI-95, Montreal, Canada, 1995

[4] Barták, R.: VisOpt – The Solver behind the User Interaction, White Paper, InSol Ltd., Israel, May 1999

[5] Barták, R.: *On-line Guide to Constraint Programming*, http://kti.mff.cuni.cz/~bartak/constraints/

[6] Bosi, F., Milano, M.: Enhancing CLP Branch and Bound Techniques for Scheduling Problems, Tech. Report DEIS-LIA-98-002, Università di Bologna, 1998

[7] Brusoni, V., Console, L., Lamma, E., Mello, P., Milano, M., Terenziani, P.: Resource-based vs. Task-based Approaches for Scheduling Problems, in: *Proceedings of the 9th ISMIS96*, LNCS Series, Springer Verlag

[8] Buzzi, S., Lamma, E., Mello, P., Milano, M.: Consistent Orderings for Constraint Satisfaction Scheduling, Tech. Report DEIS-LIA-97-001, Università di Bologna, 1997

[9] Caseau, Y., Laburthe, F.: A Constraint based approach to the RCPSP, in: *Proceedings of the CP97 Workshop on Industrial Constraint-Directed Scheduling*, Schloss Hagenberg, Austria, November 1997

[10] Caseau, Y., Laburthe, F.: Improved CLP Scheduling with Task Intervals, in: *Proceedings of ICLP94*, pp. 369-383, MIT Press, 1994

[11] Caseau, Y., Laburthe, F.: Cumulative Scheduling with Task Intervals, in: *Proceedings of JICSLP96*, pp. 363-377, MIT Press, 1996

[12] Crawford, J.M.: An Approach to Resource Constrained Project Scheduling, in: *Artificial Intelligence and Manufacturing Research Planning Workshop*, 1996

[13] Lamma, E., Mello, P., Milano, M., Temporal Constraint Handling in Scheduling Problems, Invited Paper at Intersymp95, Baden-Baden, August 1995

[14] Lever, J., Wallace, M., Richards, B.: Constraint Logic Programming for Scheduling and Planning, in *BT Technical Journal*, Vol. 13 No. 1, pp. 73-81, 1995

[15] Pegman, M.: Short Term Liquid Metal Scheduling, in *Proceedings of PAPPACT98* Conference, London, 1998

[16] Simonis, H., Cornelissens, T.: Modelling Producer/Consumer Constraints, in *Proceedings of CP95*, pp. 449-462, 1995

[17] Smith, A.W., Smith, B.M.: Constraint Programming Approaches to Scheduling Problem in Steelmaking, in *Proceedings of CP97 Workshop on Industrial Constraint-Directed Scheduling*, Schloss Hagenberg, Austria, November 1997

[18] Tsang, E.: *Foundations of Constraint Satisfaction*, Academic Press, London, 1995

[19] Wallace, M.: Applying Constraints for Scheduling, in *Constraint Programming*, Mayoh B. and Penjaak J. (Eds.), NATO ASI Series, Springer Verlag, 1994