

## CUSTOM SCHEDULING ALGORITHMS IN REAL-WORLD APPLICATIONS

Andrzej Bożek\*

\* Department of Computer and Control Engineering, Rzeszow University of Technology,  
Rzeszow, 35-959, Poland,  
Email: abozek@kia.prz.edu.pl

---

**Abstract** The paper describes the possibility of using of the Preactor Express software as a visualisation and manipulation platform for custom scheduling algorithms. The implementation of a custom scheduling algorithm for a real-world application is often far less time consuming than developing a practical and comfortable user interface for the application. It is the reason why many of scheduling algorithms remain only academic ideas. Preactor Express provides many ways of accessing scheduling data, e.g.: an interactive Gantt chart, resource calendars, a report generator. Preactor Express is free of charge software. Rules of connecting external algorithms to the Preactor Express are presented and a simple example of the connection with a custom algorithm is given in the paper. The purpose of the solution is to provide a bridge between custom innovative scheduling algorithms and a professional graphical user interface characteristic of real-world applications. It can be valuable for prototype projects for industry as well as for researchers that develop scheduling algorithms.

**Paper type:** Viewpoint  
Vol. 2, No. 3, pp. 233-245

Published online: 30 July 2012  
ISSN 2083-4942 (Print)  
ISSN 2083-4950 (Online)  
© 2012 Poznan University of Technology. All rights reserved.

**Keywords:** *production scheduling, scheduling algorithms, Preactor, user interface*

---

## 1. INTRODUCTION

Preactor Express (Preactor Homepage, 2012) is the simplest product in the family of Preactor solutions for production scheduling. It is free to use. It has been designed to allow anyone interested to test basic functionality of the Preactor software. It can be also freely used by smaller enterprises for real-world planning and scheduling if its functionality is appropriate. The set of functions that Preactor Express offers in the area of scheduling algorithms is not very extensive. However, it is possible to take scheduling data from a database by an external application, generate schedule by a custom algorithm, and save a result back to the database. Then, Preactor Express will read the prepared result and present it to a user. The paper describes how to connect a custom scheduling algorithm to Preactor Express.

The content of the paper is a partial result of the research that is being done by the author at present. The research is meant to develop a solution for flexible modelling and scheduling of production systems with the use of timed coloured Petri nets formalism (Bożek & Żabiński, 2010). The solution is going to be used in a scheduling module designed for prototype *flexible manufacturing system* (Mączka & Czech, 2010). The presented results show that it is possible to use Preactor Express as a rich graphical user interface for the scheduling module.

## 2. SCHEDULING SOFTWARE

At present, many production scheduling software packages exist. In the context of this paper, functionalities of graphical user interface (GUI) of such packages are in the focus. A GUI is a part of scheduling system which is important for all its architecture. Although this part might seem straightforward, it is indeed one of the most complex modules (Framinan & Ruiz, 2010). Information presented by a GUI can be divided into static data: plant layouts, resource calendars, routing tables, and dynamic data that represents a current schedule (Pinedo & Yen, 1997). A Gantt chart in an interactive form is most often used for dynamic data presentation and manipulation (Pinedo, 2009, pp. 358-364). Representation of resources availability is also an important issue in the practice of production scheduling. GUI components that represent resource calendars often use shift pattern convention (Dickersbach, 2009, pp. 262-264) and division into primary and secondary resources (McKay & Wiers, 2003). GUI components of scheduling systems are typically incorporated into standalone applications, but web-based solutions are also designed (Tarantilis, Kiranoudis & Thodorakopoulos, 2008).

A few commercial packages have been selected for comparison: SAP APO (Balla & Layer, 2010), Tuppas Production Scheduling Software (Tuppas Homepage, 2012), Taylor Scheduler (Taylor Homepage, 2012), JobPack Production Suite (JobPack Homepage, 2012), Preactor 500 APS and Preactor

Express (Preactor Homepage, 2012). Two academic packages have also been taken into consideration: LEKIN Scheduling System (LEKIN Homepage, 2012) and LiSA software (Andresen, Bräsel, Engelhardt & Werner, 2012).

It can be seen that the commercial packages provide at least the main GUI components such as an interactive Gantt chart, a resource calendars editor and a report generator. For example, the Gantt chart of SAP APO presents positions of activities, operations and orders over time. Additional information is also displayed, e.g. non-working times, downtimes, fixing intervals, resources and operation relationships. The Gantt interface of Tuppas software includes “what-if” tab that shows how manual changes impact performance and efficiency of the overall schedule. More details of Preactor Express GUI are described in the paragraph 3.2. Preactor 500 APS GUI has all functionality of Preactor Express GUI and additional features.

Professional paid software packages support integration with ERP/MPR/MES systems and have interfaces for data exchange with external systems. For example, SAP APO is usually linked to an ERP system, e.g. SAP R/3 or SAP ECC, which provides it with master data using the SAP APO Core Interface (CIF). JobPack can be used as standalone application or can be integrated into existing ERP/MRP system. Preactor 500 APS and other APS versions of Preactor can be integrated with ERP/MES systems using Preactor Link (Missing Link Technology, 2012).

**Table 1** Comparison of scheduling software (+ feature available, - feature unavailable, ? no direct information, n/a – not applicable)

Software feature	SAP APO	Tuppas	Taylor	JobPack	Preactor 500 APS	Preactor Express	LEKIN	LiSA
Interactive Gantt chart	+	+	+	+	+	+	+	+
Resource calendars editor	+	+	+	+	+	+	-	-
Report generator	+	?	+	+	+	+	-	-
Advanced scheduling algorithms	+	+	+	?	+	-	+	+
Support for custom routines	+	?	+	?	+	-	+	+
External data interface	+	+	+	+	+	-	+	+
Integration with ERP/MRP/MES	+	+	+	+	+	-	-	-
Demo/trial version	-	?	-	+	-	-	n/a	n/a
Express version	-	-	-	-	-	+	n/a	n/a

Commercial packages have typically advanced scheduling algorithm incorporated and they also support extending and customisation by using custom scheduling routines. For example, SAP APO has built-in heuristics for production planning, detailed scheduling, repetitive manufacturing and model mix planning. Makespan, setup times, setup costs, delay costs or mode costs (resource consumption) can be used as optimisation criteria. Taylor scheduler has a

heuristics-based engine and an algorithm-based optimiser with multi-pass optimisation strategy for schedule generation. Scheduling procedures of Preactor 500 APS include lot sizing, transfer batching, auto repeat orders, scheduling with or without material constraints, advanced material control features, resource dependant additional constraints, multiple constraints for each operation, mid-batch update and calculations, and others. It is possible to use *Preactor Events Script Processor* (PESP) or Visual Basic macros for creating custom sequencing rules.

Demo or trial versions of commercial packages are rarely available to download from websites. The academic packages (LEKIN and LiSA) are designed mainly for didactic purposes and for implementing and testing new scheduling algorithms. Although they are available as free software, they are not suitable for most practical applications, among other things because of the lack of basic GUI functionalities: a resource calendars editor and a report generator.

It is different in case of Preactor Express. This software is free of charge and free for using. Its functionality is strongly limited. Only simple automatic sequencing rules are available, that is forward, backward and bi-directional sequencing by priority, reverse priority, due date or schedule file order. However, all important components of a graphical interface (interactive Gantt chart, resource calendars editor, report generator) are present. This fact gives the possibility to use this package as GUI for external scheduling algorithms in real-world manufacturing practice. The problem is that Preactor Express has an undefined external data interface, but this limitation can be evaded and this paper presents how it can be obtained.

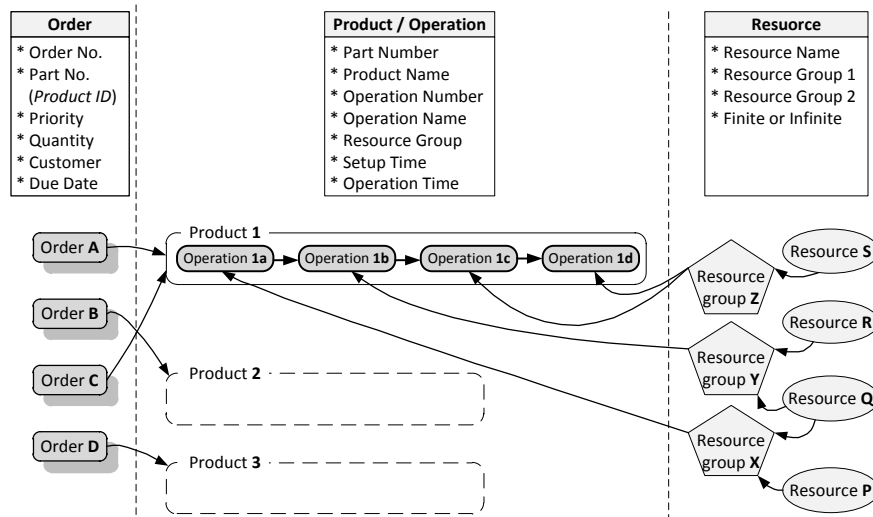
### 3. FEATURES OF PREACTOR EXPRESS

Preactor Express software will be characterised on the basis of three aspects: a scheduling data structure, a user interface, expandability and data exchange with other systems.

#### 3.1. Scheduling data structure

Preactor Express uses a scheduling data structure typical for real-world planning and scheduling systems. This data structure is based on such concepts as *order*, *product*, *operation*, *resource*, and *resource group*. The main ideas of the data organization are presented in Figure 1. Production routes are defined by sequences of operations. Definitions of all the operations are contained in one database table (or an Excel spreadsheet) and individual products are distinguished on the basis of a *part number*. Sequences of the operations are ordered by *operation numbers*. For every operation a *setup time* and an *operation time* are defined. All the operations

have a *resource group* assigned. Every *resource group* has to contain at least one *resource*. A resource can belong to one or at the most two *resource groups* simultaneously. For every resource a calendar of availability is defined. One *resource group* may be assigned to many operations in the same production route.



**Fig. 1** The scheduling data structure of Preactor Express

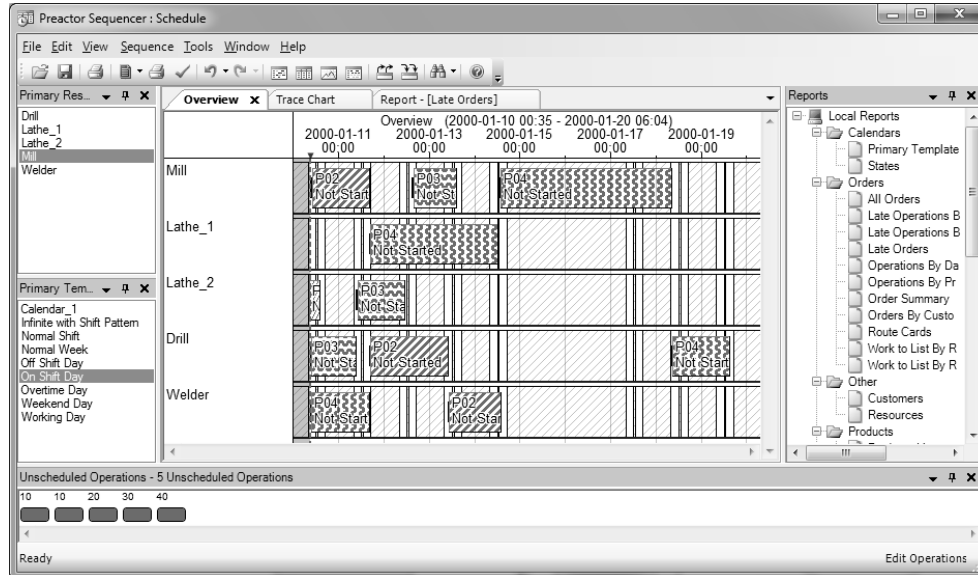
Orders define which and how many products have to be produced. It is possible to declare many orders for one product.

The scheduling data structure of Preactor Express is virtually equivalent to *flexible job shop* (FSJ) with *recirculation* and *setup times* (Pinedo, 2008, p. 15) and that is one of the most generalised representations considered in scheduling theory.

### 3.2. User interface

There are two important windows of the Preactor Express application. The main program window is the first one. It contains switched views of several panels as *Data Transfer*, *Data Maintenance*, *Configuration*, *Schedule*, and others. The *Data Maintenance* panel makes it possible to manually edit sets of resources, resource groups, products and customers.

The *Preactor Sequencer* panel is the second important window (Fig. 2). It is used for generation and manipulation of schedules. It makes also possible to edit calendars of resource availability and to prepare planning and scheduling reports that can be exported to the Excel or PDF format.



**Fig. 2** The example view of the Preactor Sequencer panel

The centre area of the *Sequencer* panel contains tabbed views of different sheets. The *Overview* is the most important sheet that represents a resource-oriented interactive Gantt chart. This chart shows sequences of operations determined by an automatic scheduling algorithm and it also makes it possible to manually move operations with a visual feedback. There are many edit fields and dialog boxes in the *Sequencer* panel that allow to modify parameters of operations, to add new calendars of availability, and so on.

### 3.3. Expandability and data exchange with other systems

Preactor Express is the software for use “as it is”, without an explicit support for expanding. It has also no support for automated data exchange with ERP/MRP and MES systems.

The scheduling data transfer can be carried out with the use of Excel spreadsheets or a database. Excel spreadsheets have to have a predefined structure of columns. Only definitions of orders, products and resources can be transferred between spreadsheets and Preactor Express. Preactor stores all scheduling and configuration data in a database. All data can be accessed by an external application using this database. However, Preactor Express imposes some limitations of the use of the database:

- The structure of database tables is defined arbitrary and cannot be altered.
- A user instance of *SQL Server* (Wolter, 2005) database account has to be used.

#### 4. CONNECTING EXTERNAL APPLICATION TO PREACTOR EXPRESS

As it has been stated before, Preactor Express has the attractive graphical user interface with the complex functionality useful in real-world planning and scheduling. Thus, it can be very profitable to plug an external advanced scheduling algorithm into this software.

##### 4.1. Data flow in Preactor Express

The simplified diagram of data flow in Preactor Express is presented in Figure 3. The database is the central part of the data flow mechanism in the application. Data can be accessed in several ways. It is possible to import some elements of data (resources, products and orders) from Excel spreadsheets (Fig. 3). A planist can control all Preactor data using *Data Maintenance* and *Preactor Sequencer* panels. All data can be also accessed by an external application using a direct connection to the database.

Any time, when *Preactor Sequencer* starts or an existing schedule is opening, the scheduling data is loaded from the database. When a user saves a schedule, the scheduling results are copied from *Sequencer* to the database. A custom scheduling algorithm has to be connected directly to the database, because Preactor Express does not support other forms of connecting.

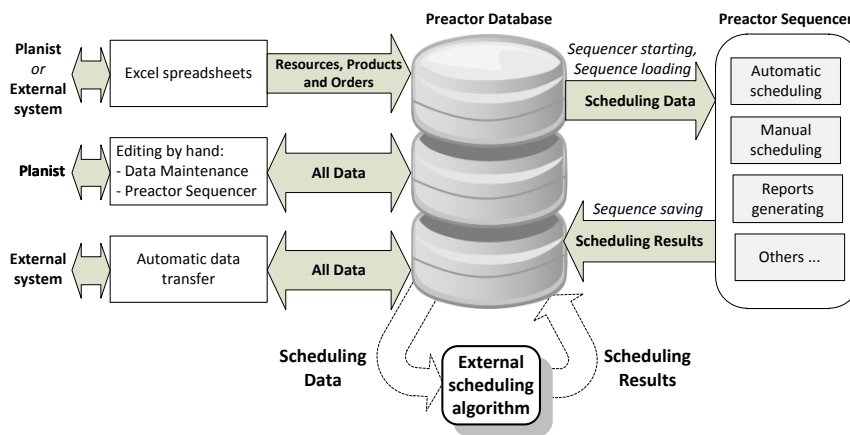


Fig. 3 The simplified diagram of data flow in Preactor Express

An external application reads data from the database, executes a custom scheduling algorithm, and saves scheduling results back to the database (Fig. 3). It is one peculiar feature of Preactor system that makes that connection reasonable. When data is loaded from the database to *Preactor Sequencer*, there is no

verification or rescheduling process. Data is loaded directly from database tables. As a result, an external scheduling algorithm can be applied in two steps:

1. The external algorithm is executed on demand of a planist and a result is saved in the Preactor database.
2. Preactor Sequencer is opened or the schedule is reloaded with the new content that represents the sequence just generated by the custom algorithm.

## 4.2. Database structure of Preactor Express

The structure of the Preactor Express database contains 5 schemas. All data, important for the connection between Preactor and a custom algorithm, is contained in the tables from the *UserData* and *Calendar* schemas. The tables from the *UserData* schema store a set of data that represents a scheduling problem (orders, products, customer, resources, resources group) and scheduling results. The *Calendar* schema groups tables that define calendars of resources availability.

The *Products* table describes all products and operations of a scheduling problem. Every operation is represented by one row in the table. The *ResourceGroupResources* table assigns a resource identifier to a resource group identifier for every resource. The *Orders* table contains all orders that have to be scheduled. Similarly as in the *Products* table, every row represents one operation.

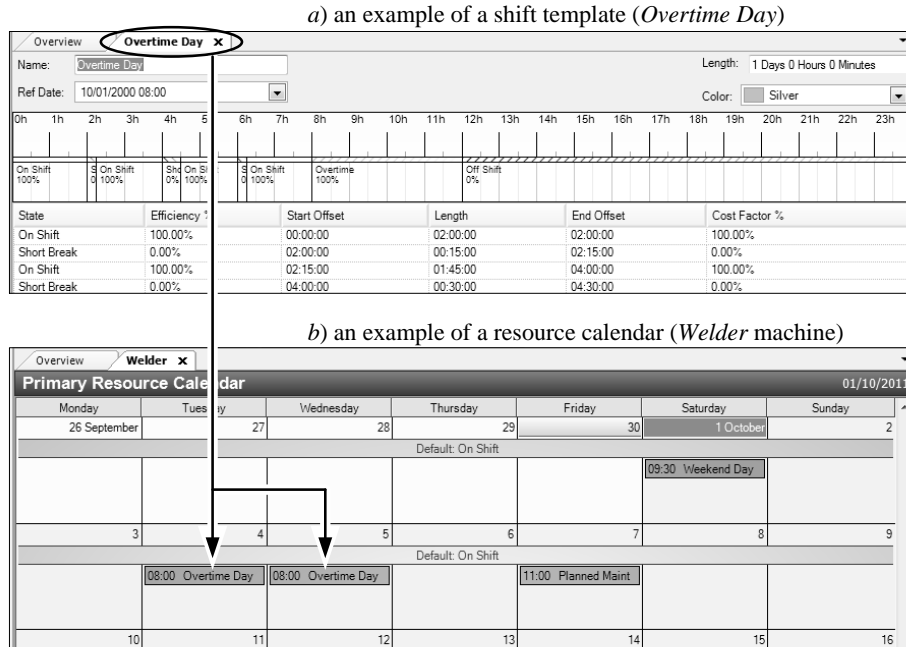


Fig. 4 Definition of resource availability in Preactor Express



It is important that *Orders* table stores data that is used by *Preactor Sequencer* for creating Gantt charts and reports. Thus, scheduling results of a custom algorithm have to be written only to the *Orders* table. There are three fields in the *Orders* table that are especially important for the scheduling result representation: *SetupStart*, *StartTime* and *EndTime*.

Resource availability is defined by resource calendars. The calendars assign shift templates to selected periods of time (Fig. 4). All data that defines resource availability is also stored in database (5 tables) and it can be accessed by an external application.

## 5. CUSTOM ALGORITHM – EXAMPLE

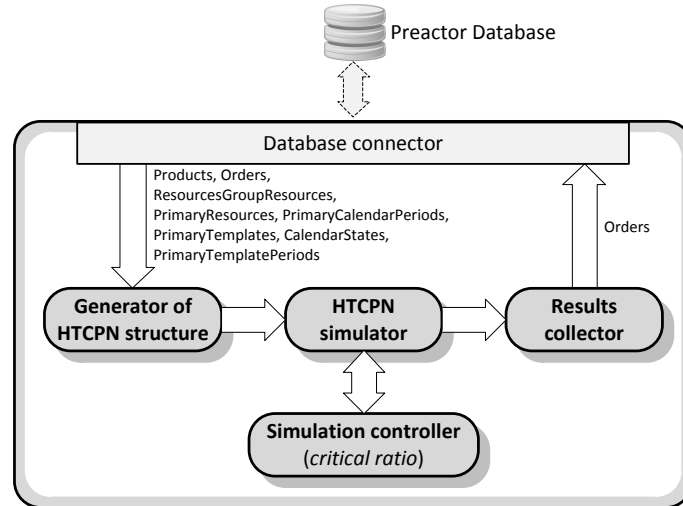
For testing, an external module has been prepared that performs a custom scheduling algorithm. This algorithm is based on representation of a production system by *Hierarchical Timed Coloured Petri Net* (HTCPN) (Jensen & Kristensen, 2009). More details of the algorithm are presented in the works (Zhang, Gu & Song, 2008; Božek & Żabiński, 2010; Božek, 2012).

### 5.1. Implementation of custom algorithm

The simplified block diagram of the external scheduling module is shown in Figure 5. This module has been coded in the Java programming language. The application takes data from the Preactor database, generates a new schedule and saves results back to the database. It is performed in several steps:

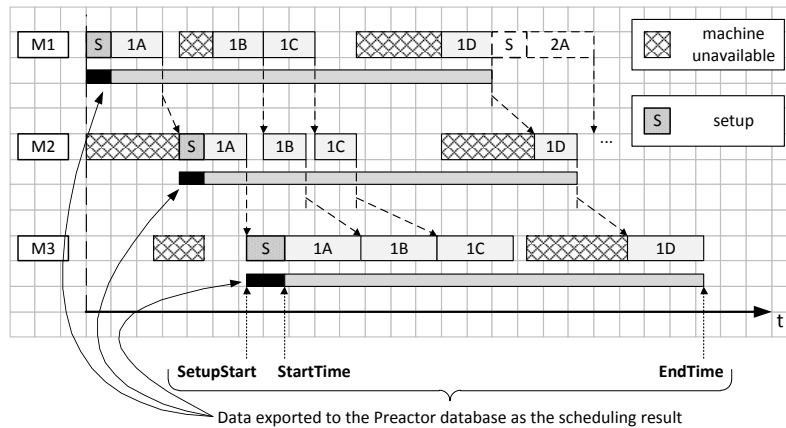
- A content of a few tables is converted to a HTCPN net model by the *generator of HTCPN structure* block. The structure of the net is inferred from the tables *Product*, *Orders* and *ResourcesGroupResources*. Machine calendars are defined on the basis of the tables *PrimaryResources*, *PrimaryCalendarPeriods*, *PrimaryTemplatePeriods*, *PrimaryTemplates* and *CalendarStates*.
- The net model is simulated by the *HTCPN simulator* block. When a conflict of operations occurs the *simulation controller* selects an operation for execution on the basis of minimal *critical ratio* value.
- Results of the simulation, collected by the *results collector* block, are converted to the form proper for Preactor and saved in the database (the table *Orders*).

The custom algorithm based on HTCPN model supports the scheduling with transfer batching. The sequencing procedures of the Preactor Express do not support this functionality, quantity of an order is used only for determining total time of an operation.



**Fig. 5** The block diagram of the external scheduling module

In the custom algorithm, a unit of the quantity is regarded as a transfer batch. Any batch can be transferred independently from one operation to the consecutive one. An example of a simulation of such production with transfer batching is shown in Figure 6.



**Fig. 6** The simulation of the production with transfer batching

Parameters of every operation are extracted from the simulation results (Fig. 6):

- *SetupTime* – the moment, when a setup for the operation starts,
- *StartTime* – the moment, when processing of the first batch of the operation starts,
- *EndTime* – the moment, when processing of the last batch of the operation ends.

These parameters are registered during the simulation by the *results collector* block and sent to the Preactor database as the scheduling result.

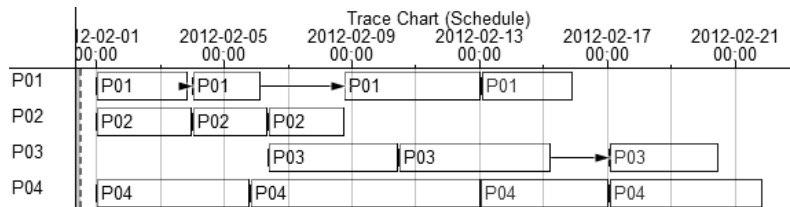
### 5.2. Verification of solution

An example scheduling problem has been defined to verify the possibility of connection between Preactor and the external scheduling algorithm. There are 5 machines, 4 products and 4 orders, one order for every product.

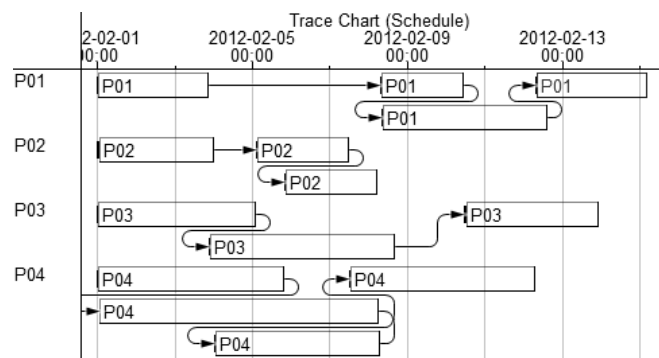
**Table 2** The example scheduling problem - orders

Order Name	Product Name	Number of Operations	Priority	Quantity	Due Date
P01	Connector	4	30	205	2012-02-13
P02	Handle	3	20	170	2012-02-08
P03	Bracket	3	40	195	2012-02-18
P04	Piston	4	10	230	2012-02-14

The set of orders is described in Table 2. Every order has a multiple quantity, so the functionality of transfer batching is relevant. The scheduling result is presented in the form of a task-oriented Gantt chart in Figure 7. It has been obtained by the use of the standard built-in Preactor Express option *Forward by Due Date*.



**Fig. 7** The schedule generated by the Preactor Express built-in algorithm



**Fig. 8** The schedule generated by the external custom scheduling algorithm

The schedule generated for the same scheduling problem by the custom algorithm is shown in Figure 8. Due to the transfer batching, some of the operations overlap. Preactor presents the results generated by the external algorithm in a very legible form, even though Preactor Express itself has no scheduling rules leading to overlapping of operations.

## 6. CONCLUSION

The concept of the connection of an external scheduling algorithm to Preactor Express software has been analysed. Preactor Express is characterised by very rich graphical user interface. It makes possible to represent schedules in several forms, to modify schedules manually, to generate valuable reports, to edit calendars of resources, etc. However, the set of scheduling procedures is strongly limited. On the other hand, there are often available sophisticated scheduling algorithms, but they are not used because of the lack of a graphical interface. Connection of these algorithms to Preactor Express may be a good solution.

The use of Preactor Express as a schedule presentation and manipulation platform can be valuable for prototype projects for industry. It can be also useful for researchers that develop scheduling algorithms. They received a convenient environment for presentation of results. Developing of an algorithm for connecting with Preactor is also some kind of a challenge, because that algorithm has to conform to complex scheduling data, typical of real-world problems.

## REFERENCES

- Andresen M., Bräsel H., Engelhardt F., Werner F., (2012), "LiSA – A Library of Scheduling Algorithms. Handbook for version 3.0", available at: <http://lisa.math.uni-magdeburg.de/download/lisa-3.0-manual-english.pdf> (accessed 10 January 2012).
- Balla J., Layer F., (2010), Production planning with SAP APO. SAP Press, Galileo Press GmbH, Bonn.
- Bożek A., Żabiński T., (2010), "Colored timed Petri Nets as tool of off-line simulating for intelligent manufacturing systems", [in:] *Przegląd Elektrotechniczny (Electrical Review)*, Vol.2010, No.9, pp. 101-105.
- Bożek A., (2012), "Using Timed Coloured Petri Nets for Modelling, Simulation and Scheduling of Production Systems", Righi R. (Ed.), *Production Scheduling*, InTech, Rijeka (Croatia), pp. 207-230.
- Dickersbach J. T., (2009), *Supply Chain Management with APO*, Springer, Berlin Heidelberg.
- Framinan J. M., Ruiz R., (2010), "Architecture of manufacturing scheduling systems: Literature review and an integrated proposal", [in:] *European Journal of Operational Research*, Vol.205, No.2, pp. 237-246.

- Jensen K., Kristensen M. L., (2009), *Coloured Petri Nets : Modelling and Validation of Concurrent Systems*, Springer-Verlag, Berlin Heidelberg.
- JobPack Homepage, (2012), JobPack Production Suite, available at: <http://www.jobpack.com> (accessed 10 January 2012).
- LEKIN Homepage, (2012), LEKIN – Flexible Job-Shop Scheduling System, available at: <http://community.stern.nyu.edu/om/software/lekin> (accessed 10 January 2012).
- Mączka T., Czech T., (2010), “Manufacturing Control and Monitoring System – Concept and Implementation”, *IEEE International Symposium on Industrial Electronics, Bari (Italy)*, Vol.1, pp. 3900-3905.
- McKay K. N., Wiers V. C. S., (2003), “Integrated decision support for planning, scheduling, and dispatching tasks in a focused factory”, [in:] *Computers in Industry*, Vol.50, No.1, pp. 5-14.
- Missing Link Technology, (2012), Connectelligence, Preactor Link, available at: <http://www.missinglinktechnology.com/Connectelligence/PreactorLink/tabid/104/Default.aspx> (accessed 10 January 2012).
- Pinedo M. L., Yen B. P.-C., (1997), “On the design and development of object-oriented scheduling systems”, [in:] *Annals of Operation Research*, Vol.70, No.0, pp. 359-378.
- Pinedo M. L., (2008), *Scheduling: Theory, Algorithms, and Systems*, Springer, New York.
- Pinedo M. L., (2009), *Planning and Scheduling in Manufacturing and Services*, Springer, New York.
- Preactor Homepage, (2012), available at: <http://www.preactor.com> (accessed 10 January 2012).
- Tarantilis C. D., Kiranoudis C. T., Thoedorakopoulos N. D., (2008), “A WEB-based ERP system for business services and supply chain management: Application to real-world process scheduling”, [in:] *European Journal of Operational Research*, Vol.187, No.3, pp. 1310-1326.
- Taylor Homepage, (2012), Taylor Scheduler, available at: [http://www.taylor.com/products/taylor\\_scheduler.php](http://www.taylor.com/products/taylor_scheduler.php) (accessed 10 January 2012).
- Tuppas Homepage, (2012), Tuppas Production Scheduling Software, available at: <http://www.tuppas.com/Production%20Scheduling.htm> (accessed 10 January 2012).
- Wolter R., (2005), „MSDN Library: SQL Server 2005 Express Edition User Instances”, available at: [http://msdn.microsoft.com/en-us/library/bb264564\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/bb264564(v=sql.90).aspx) (accessed 25 July 2011).
- Zhang H., Gu M., Song X., (2008), „Modeling and Analysis of Real-life Job Shop Scheduling Problems by Petri nets”, *41st Annual Simulation Symposium (anss-41 2008)*, Ottawa, pp. 279-285.

## BIOGRAPHICAL NOTES

**Andrzej Bożek** is a teaching-research assistant in the Department of Computer and Control Engineering at the Rzeszow University of Technology. His research focuses on modelling and simulation of production systems, especially with the use of the Petri net formalism. His scientific interests also include design and implementation of production planning, scheduling and control solutions for real-world applications.

