Simulation in Produktion
und Logistik 2017
Sigrid Wenzel & Tim Peter (Hrsg.)
kassel university press, Kassel 2017

Improving the Accuracy of Cycle Time Estimation for Simulation in Volatile Manufacturing Execution Environments

Verbesserung der Schätzgenauigkeit von Zykluszeiten für die Simulation von volatilen Fertigungsumgebungen

András Pfeiffer, Dávid Gyulai, László Monostori, Institute of Computer Science and Control, Hungarian Academy of Sciences, Budapest (Hungary),
pfeiffer.andras@sztaki.mta.hu, gyulai.david@sztaki.mta.hu,
monostori.laszlo@sztaki.mta.hu

Abstract: The paper introduces a decision support architecture with an integrated, self-building simulation module for the validation of the calculated manufacturing capacities and analysis of the effect of possible actions in volatile manufacturing environments. The paper addresses the simulation module in which an effective filtering algorithm is proposed for cleaning cycle time log data provided by a Manufacturing Execution System (MES), as necessary input for simulating manufacturing processes. The main functionalities and advantages are highlighted by a real industrial case study.

1 Problem Formulation

At the operational level of manufacturing systems difficulties arise from unexpected tasks and events, non-linearity, and a multitude of interactions while attempting to control various activities in dynamic shop floors. The selection of the most appropriate production control decision for a given assignment, as well as the prediction of waiting times, workloads or utilizations of the resources are not trivial tasks, although they can be supported by simulation-based evaluations (Pfeiffer et al. 2007; Bagchi et al. 2008; Watt 1998).

Therefore, based on previous results (Pfeiffer et al. 2016; Pfeiffer et al. 2011; Monostori et al. 2007), we propose a decision support architecture, in which the integrated, self-building simulation module can be applied for validation of the calculated manufacturing capacities, a-priori recognition of due date deviations and analysis of the effect of possible actions taken. In the research presented in this paper a special emphasis is given to the prediction and evaluation of the production on a daily, rolling time horizon (e.g., work in process (WIP) trajectories, machine utilizations).

An important issue regarding short-term (operational level) simulation is the automatic collection and definition of simulation input data. Therefore, as the main focus of the proposed paper a new cycle time definition method (filtering method) is presented, as well as a self-building simulation procedure is described in details.

2 Simulation-supported Production Control

2.1 Literature Review

The discrete-event simulation (hereafter referred to as simulation) approach has been applied to decisions in scheduling and control, related to production applications (Banks 1998; Law and Kelton 2000; O'Rielly and Lilegdon 1999). The simulation models that are used for making or evaluating these decisions (e.g., by projecting different key performance indicators, KPIs) generally represent the flow of materials to and from processing machines and the operations of machines themselves (Rabelo et al. 2003). Potential problems can be identified and can be corrected using a simulation model. By far the most common use of simulation models is for operational decisions such as scheduling or dispatching (Law and Kelton 2000).

As it is presented in the related works describing some of the application areas, as well as the recent solutions of simulation in production control, simulation has been typically used for off-line decision making. Consequently, effective integration into the control process of production was restrained. One of the limitations of its use in on-line decision making is the considerable amount of time spent in gathering and analysing data. In quasi real-time control (hours, minutes), however, the three key issues are data acquisition, quick response and instantaneous feed-back. As a result, decision makers mostly apply simulation primarily for off-line decision support and not for the critical on-line decision making that may arise.

2.2 Proposed Simulation-based Prediction Framework

The main goal of the framework to be introduced here is to provide a self-building production simulation, capable of both prospective (e.g. locate anticipated disturbances, identify trends of designated performance measures), and reactive (e.g. gathering statistics on resources) simulation functionalities. Self-building simulation means that the simulation model is built up by means of the combination of the MES data as well as the knowledge extracted from the MES data (e.g. resource and execution model). In addition to the automatic model building feature, main requirement of the solution is to minimize the response time of the experiments and to enable the quasi "real-time" applicability of the simulation (Pfeiffer et al. 2016; Váncza et al. 2011).

Regarding the main operation modes of the simulator in the proposed architecture (Fig. 1) are as follows:

- Off-line validation, sensitivity analysis and statistical modelling of the system. Evaluation of the robustness of the system against uncertainties (e.g., different control settings, thresholds and system load levels). Consequently, this scenario analysis might point out the resources or settings which can endanger the normal operation conditions. In Figure 1 off-line simulation represents the comprehensive model of the plant.

- On-line, anticipatory recognition of deviations from the planned operation conditions by running the simulation parallel to the plant activities; and by using a look ahead function, support of situation recognition (proactive operation mode, Fig. 1).
- On-line analysis of the possible actions and minimization of the losses after a disturbance already occurred (reactive operation mode, Fig. 1), e.g., what-if scenario analysis.

In Figure 1, Plant represents the underlying production system, which is generally controlled through the MES. Thus, green arrows represent production related data provided by the plant (e.g., resource status, job completion, or, the performance measure of interest KPI in the current case), either gathered by the MES and stored as log data, or, monitored on-line by, i.e. the simulation framework. Contrary, grey arrows represent an interaction or information exchange, e.g., the Decision-maker might control the process of the production (highlighted as Reaction) of the plant by the MES system.

In a real-world application, the three main distinct operation modes follow each other during operation.

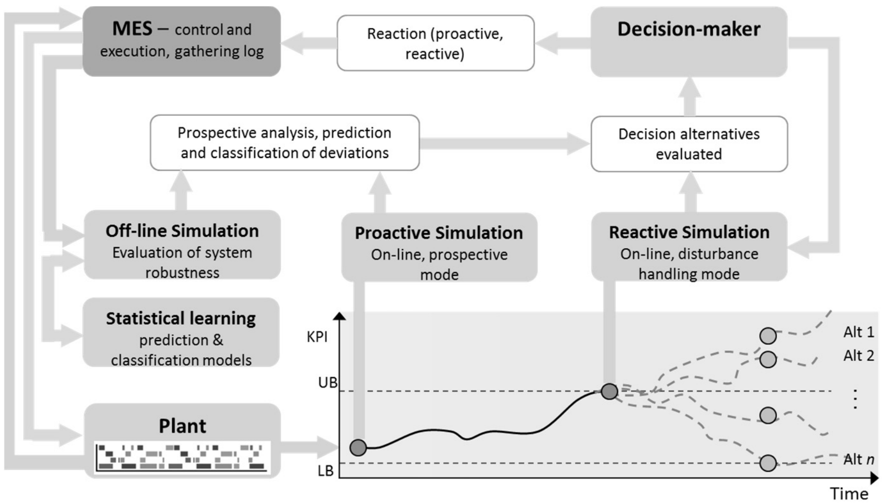

Figure 1: *Plant-level active disturbance handling realized by using statistical learning methods and reactive/proactive operation modes for simulation*

In contrast to the on-line proactive mode of the simulation, in the off-line scenario, simulation is applied in combination with the MES log data for setting up and parameterizing statistical learning prediction models (Statistical learning, Prediction and classification models in Fig. 1). Once the prediction models are set as an off-line analysis, permanent, on-line simulation analysis of the manufacturing system is performed (highlighted by Time in the bottom right corner of Fig. 1). This means a rolling horizon monitoring of the productions systems' behaviour (e.g. by monitoring preselected performance measure of interest, e.g. LT of jobs) in advance by using prospective simulations. In case of a relevant deviation occurs, i.e., a situation is recognized which might endanger the production, a prospective analysis and classification of the deviations are performed. At this point the models obtained in the previous, off-line mode are combined with the actual simulation results in order to analyze the possible effect of the deviation, and moreover, to filter out unnecessary interventions. For instance, in Figure 1 LT is expected to be out of the range defined by upper bound (UB). Consequently, reactive simulation mode is initiated, where a predefined set of possible solutions (Decision alternative 1 – Decision alternative n, denoted as, e.g., Alt 1) for normalizing the production is preformed, highlighted as disturbance handling mode in Figure 1.

2.3 Defining Operation Cycle Times on MES Data

In simulation systems representing a complex, wide scale production system, exact processing times are crucial for successful and credible simulation results.

Defining process times for simulation models based on logged production (e.g. MES) data is a well-known and widely used method. Bagchi et al. (2008) present a linear regression method for calculating process times based on raw process times (RPT) collected for single, batch and sampling tool types. Sivakumar and Chong (2001) present a case study, where the theoretical process times are defined by the mean values, however the authors state that based on the wide distribution in theoretical process times, theoretical ratio based on mean is indefinite.

The main goal is to define the so called basic process time for the given operation and machine (or tool) relations, i.e., to find for each case the real, technologically acceptable minimum cycle times by applying the filtering algorithm proposed. Since several factors influence the raw process time as for instance different waiting times in the input buffer of machines, different operators, etc., the most relevant way is to scrape the raw process times from the effect of disturbing factors, and thus, to identify the relevant lower bound (excepting unnecessary, problematic data), i.e., the shortest possible raw process time of the data. It is clear that by calculating the mean or the mode values for the data set does not provide the necessary lower bound.

3 Proposed Methods and Preliminary Simulation Results

As one of the most important issues regarding the self-building simulation system, a significantly more effective method (compared to e.g. mean or modus statistical methods) had to be developed in order to have reliable and up-to date cycle times in the simulation.

The paper focuses on the novel method developed for filtering raw processing time data for cycle time calculation, and on applying it for decision support based on the proposed self-building simulation tool. The effectiveness of the methods is presented through computational experiments on data provided by a real industry case study.

3.1 Filtering MES Log Data

In order to extend the capabilities of the simulation towards prediction and estimation of future scenarios' results, the use of statistical learning models is proposed.

Basically, statistical learning refers to a set of methods for understanding and learning from data and providing solutions to understand the correlations among parameters and processes James et al. (2013). Thus, as one of the most important issues regarding the self-building simulation system, a significantly more effective method (compared to e.g., mean or modus statistical methods) had to be developed in order to have reliable and up-to date cycle times in the simulation.

3.1.1 Cycle Time Definition Algorithm

Contrary to several cases shown in the literature review, here the raw cycle times for single-capacity and multi-capacity (pipeline) machines cannot be defined directly based on the logged data, i.e., cannot be calculated as the difference between claim-out and claim-in timestamps (see Fig. 2).

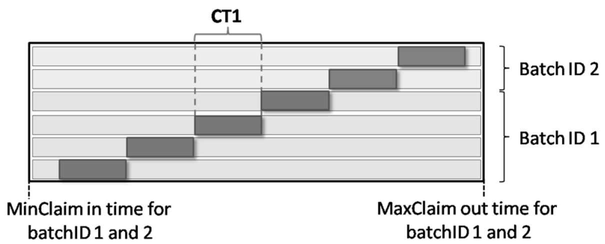

Figure 2: *Minimum and maximum claim-in and claim-out times for a lot containing six products in two different batches, logged on a selected single-capacity machine*

Due to the distinctiveness of the claiming procedure, available raw cycle times constantly contain waiting times in front of and after the tools, moreover, those lots where the number of products are greater than 1 (e.g. it equals six in the example given in Fig. 2), the raw cycle time contains the processing time (for single) or cycle and tact (for multi) time for all the products in the lot (raw cycle time is the difference between MaxClaim-out time and MinClaim-in time). Regarding batch tools, the raw cycle time calculated with the above method contains the waiting times only. Therefore, the problem for cycle time definition is to first create raw basic cycle times calculated for one product (CT1, Fig. 2) and second, to calculate the relevant lower bound for each existing distinct operation / product type pairs. Calculated CT1 values are stored in the MES database and used by the simulation for calculating lot cycle times, described in details in Section 2.2. Instead of artificial groups, batches are considered as the basis of the cycle time definition (1).

$$CT1_i = \frac{maxT_{i,out} - minT_{i,in}}{n_i} \tag{1}$$

where $CT1_i$ is the basic cycle time calculated for batch *i,* and n_i is the number of parts assigned to batch *i.*

The following algorithm describes the cycle time calculation method developed:

1. Get all product type /operation type pairs from MES Log data (e.g., 80,000 exists in the case-study presented).

2. For a defined pair, basic cycle times (CT1) are calculated based on the MES log data, regarding each batch in consideration (e.g., for more than 50 logs available in the MES Log database).
3. For the defined pair the type of tools (single, multi, batch) on which the operation can be processed are defined.
4. Classification algorithm is applied to identify the relevant minimum value for CT1.
5. If further product type/operation type pair exist then select next pair and return to 2.
6. Stop.

Figure 3 presents the calculated CT1 (solid red line at 528 sec) for a given operation type and product type pair. Raw cycle time values are coloured by the product subtypes. The mean value is 1,920 seconds, which is more than the triple of the CT1 value calculated by the new method, though, an upper bound has been applied for eliminating extremely extraneous data.

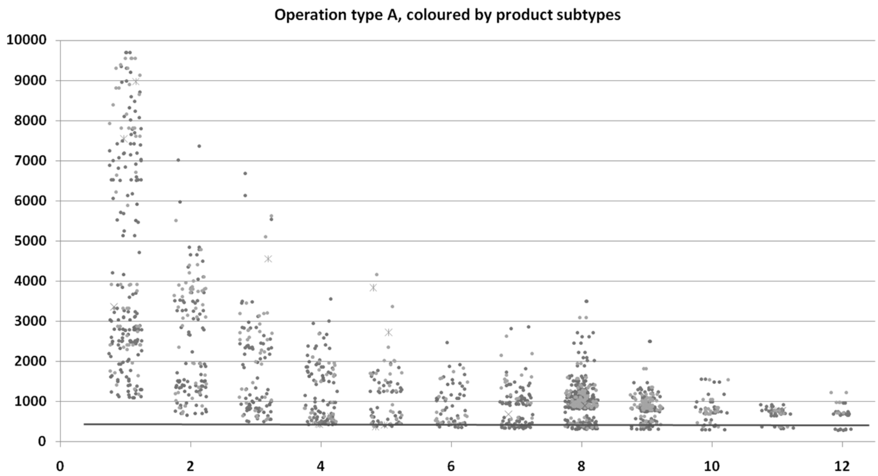

Figure 3: *Resulted CT1 for a selected operation type and product type pair (on the X-axis the number of parts in one batch, while on the Y-axis the cycle times in seconds are highlighted)*

3.1.2 Classification Algorithm for CT1

In the previous section, the main algorithm of the developed cycle time definition method was described, which needs further specification in order to be able to recognize the findings regarding the calculation of the lower bound of raw CT values.

All the calculated raw cycle times are grouped into classes and unnecessary data is removed, according to the following method:

1. Define the number of classes. The required number of classes (k) is defined by the Sturges' rule (Hyndman 1995) $k = 1 + \log_2 n$, where n is the number of observations.
2. Define class bounds. This is calculated by dividing the range (upper bound, UB) of the observed values to k classes. Class width (CW) is then equals to UB/k.

3. Cut unnecessary data both from left and right side (typically the 10 % of total „weight" can be neglected). For instance, in top and middle charts in Figure 4, values in the classes marked with red circles are removed from the data set (i.e., class bound are set so that these data will be set out of the range).
4. If no more data can be neglected continue, else return to 2.
5. Define maximum derivative between classes (e.g. red arrow pointing at class 3 in the bottom chart in Figure 4.)

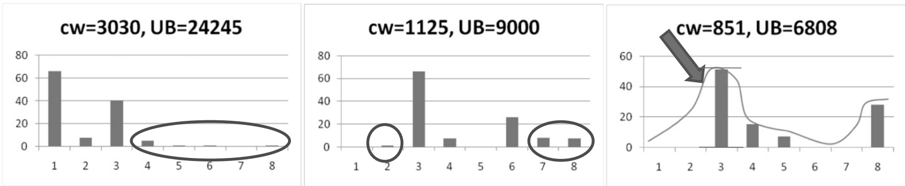

Figure 4: *Example of three different stages of the classification algorithm*

CT1 equals the mean of the lower and upper class bounds. The method presented here will find the steepest part of an imaginational curve fitted to the classified CT1 values. Great potential can be found in the possible improvement of the classification algorithm. For instance, increasing the number of classes after the last point reached in the algorithm, or sweeping the class bounds around a selected area are both reasonable solutions for the development.

3.2 Computational Experiments

3.2.1 Cleaning Raw Data

The methods introduced in the paper has been tested on a large-scale industrial data set. The methods in the prototype system has been implemented in R.

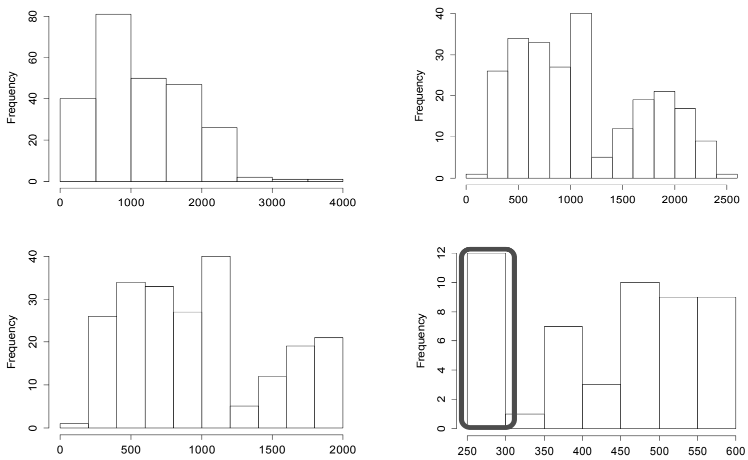

Figure 5: *Histograms for resource ID 42 (with starting limit 4000), 1st, 3rd, 5th, and 15th iteration (X-axis represents time limit values, while Y-axis is the frequency)*

The system cleans raw MES log data during the test procedure provided here by affording cycle time definition for selected machines, operations and product type triplets. The test data set for the analysis contains more than 50 k log entries for 13 variables (note that the real system provides 10 k entries daily).

As an illustrative example of the results to be presented, in Figure 5 a typical case is highlighted for a selected machine. The original range for the cycle time has been reduced to 250-300.

3.2.2 Validation – Comparison of Throughput

In order to have credible results computed by the simulation model, a comprehensive validation procedure is mandatory. In the followings, a short example is given about the validation of throughput values.

One stream of the validation procedure of the proposed system is based on the comparison of the simulated WIP prediction results (simulated log) with the real original data (Fig. 6). This validation procedure serves as a feed-back for the iterative fine-tuning (trace technique, Law and Kelton (2000)) of the execution policies, tool models, process times and dispatching rules applied in the simulation.

In Figure 6 the results of a comparison for one-week period is highlighted. If the total number of products in the system (WIP) is considered as 100 %, then the ratio of exiting products all together in the simulation for the one-week time horizon in consideration is 1.222 %, while the real, log based ratio is 1.306 %. The difference between the predicted and the real total throughput is relatively low (6.38 %), however the distinct steps in the log based curve cannot be represented exactly with the simulation.

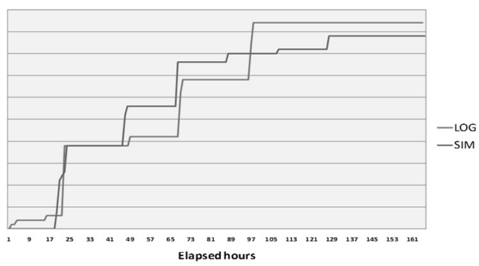

Figure 6: *Simulated and real throughput diagram, noted as SIM and LOG. X-axis represents the elapsed hours, while Y-axis the cumulated number of products exiting.*

4 Conclusions

The paper introduced a decision support architecture with an integrated, self-building simulation module for the validation of the calculated manufacturing capacities, and analysis of the effect of possible actions in volatile manufacturing environment. The paper addressed the simulation module in which an effective filtering algorithm is proposed for cleaning cycle time log data provided by a MES, as necessary input for simulating manufacturing processes. A novel cycle time definition algorithm was proposed for raw basic cycle time calculation. However, further filtering of raw cycle times was needed, thus a new method, based on classification, defining lower bounds for effective cycle times were provided. The main functionalities and advantages were highlighted through a real industrial case study and validated by simulation experiments. Further research work has to be taken on applying the above introduced methods on an online, close-loop manner, making short-term simulations for decision support more effective.

Acknowledgements

The work for this paper was supported by the European Commission through the H2020 project Centre of Excellence in Production Informatics and Control (EPIC, Ref. No.: 739592) and by the GINOP-2.3.2-15-2016-00002 grant.

References

Bagchi, S.; Chen-Ritzo, C.; Shikalgar, S.T.; Toner, M.: A full-factory simulator as a daily decision-support tool for 300mm wafer fabrication productivity. In: Mason, S.J.; Hill, R.R.; Mönch, L.; Rose, O.; Jefferson, T.; Fowler, J.W. (eds.): Proceedings of the 2008 Winter Simulation Conference. 2009, pp. 2021-2029.

Banks, J.: Handbook of Simulation, Principles, Methodology, Advances, Application and Practice. John Wiley & Sons Inc 1998.

Cowling, P.; Johansson, M.: Using real time information for effective dynamic scheduling. European Journal of Operational Research 139 (2002), pp. 230-244.

Honkomp, S.J.; Mockus, L.; Reklaitis, G.V.: A framework for schedule evaluation with processing uncertainty. Comp. and Chem. Eng. 23 (1999), pp. 595-609.

Hyndman, R.J.: The problem with Sturges' rule for constructing histograms. Department of Econometrics and Business Statistics, Monash University, Clayton, Victoria, Australia, 1995, pp. 1-2.

James, G.; Witten, D.; Hastie, T.; Tibshirani, R.: An Introduction to Statistical Learning. New York: Springer 2013.

Kim, Y.; Lee, D.; Kim, J.; Roh, H.: A simulation study on lot release control, mask scheduling, batch scheduling in semiconductor wafer fab facilities. Journal of Manufacturing Systems 17 (1998) 2, pp. 107-117.

Klein, M.; Kalir, A.: Improved simple simulation models for semiconductor wafer factories. In: Perrone, L.F.; Wieland, F.P.; Liu, J.; Lawson, B.G.; Nicol, D.M.; Fujimoto, R.M. (eds.): Proceedings of the 2006 Winter Simulation Conference, 2006, pp. 1836-1839.

Law, A.; Kelton, D.: Simulation modeling and analysis. New York: McGraw-Hill 2000.

Lin, J.T.; Wang, F.; Yen, P.: Simulation analysis of dispatching rules for an automated interbay material handling system in wafer fab. International Journal of Production Research 39 (2001) 6, pp. 1221-1238.

Monostori, L.; Kádár, B.; Pfeiffer, A.; Karnok, D.: Solution approaches. To real-time control of customized mass production. CIRP Annals - Manufacturing Technology 56 (2007) 1, pp. 431-434.

O'Reilly, J.J.; Lilegdon, W.R.: Introduction to FACTOR/AIM. In: Proceedings of the 1999 Winter Simulation Conference, 1999, pp. 201-207.

Pfeiffer, A.: Novel Methods for Decision Support in Production Planning and Control, Thesis (PhD). Budapest University of Technology and Economics, 2007.

Pfeiffer, A.; Kádár, B.; Monostori, L.; Vén, Z.; Popovics, G.: Co-analysing situations and production control rules in a large-scale manufacturing environment. In: 44th CIRP Int. Conf. on Manufacturing Systems – New Worlds of Manufacturing, Wisconsin, Madison (USA), 2011, paper No. 101.

Pfeiffer, A.; Gyulai, D.; Kadar, B.; Monostori, L.: Manufacturing lead time estimation with the combination of simulation and statistical learning methods. PROCEDIA CIRP 41 (2016), pp. 75-80.

Rabelo, L.; Helal, M.; Jones, A.; Min, J.; Son, Y.J.; Deshmukh, A.: A hybrid approach to manufacturing enterprise simulation. In: Proc. of the 2003 Winter Simulation Conference, 2003, pp. 1125-1133.

Sabuncuoglu, I.; Kizilisik, O.M.: Reactive scheduling in a dynamic and stochastic FMS environment. Int. J. of Production Res. 41 (2003) 17, pp. 4211-4231.

Sivakumar, A.I.; Chong, C.S.: A simulation based analysis of cycle time distribution, and throughput in semiconductor backend manufacturing. Computers in Industry 45 (2001) 1, pp. 59-78.

Smith, S.F.: Knowledge-based production management: Approaches, results and prospects. Production Planning & Control 3 (1992) 4, pp. 350-380.

Váncza, J.; Monostori, L.; Lutters, E.; Kumara, S.R.; Tseng, M.; Valckenaers, P.; Van Brussel, H.: Cooperative and Responsive Manuf. Enterprises, CIRP Annals–Manuf. Technology 60 (2011) 2, pp. 797-820.

Vieira, G.E.; Herrmann, J.W.; Lin, E.: Predicting the performance of rescheduling strategies for parallel machines systems. Journal of Manufacturing Systems 19 (2000), pp. 256-266.

Watt, D.G.: Integrating simulation-based scheduling with MES in a semi-conductor fab. In: Benjamin et al. (eds.): Proceedings of the 1998 Winter Simulation Conference, 1998, pp. 1713-1715.

Wilhelm, R., et al.: Finite Schedule Monitoring and Filtering in a Computer Integrated Manufacturing Environment. Annals of the CIRP 49 (2000) 1, pp. 335-338.