Event Based Identification and Prediction of Congestions in Manufacturing Plants

Clemens Schwenke, Thomas Wagner, Klaus Kabitzsch

To cite this version:

Clemens Schwenke, Thomas Wagner, Klaus Kabitzsch. Event Based Identification and Prediction of Congestions in Manufacturing Plants. George L. Kovács; Detlef Kochan. 6th Programming Languages for Manufacturing (PROLAMAT), Oct 2013, Dresden, Germany. Springer, IFIP Advances in Information and Communication Technology, AICT-411, pp.376-390, 2013, Digital Product and Process Development Systems. <10.1007/978-3-642-41329-2_37>. <hal-01485832>

HAL Id: hal-01485832

https://hal.inria.fr/hal-01485832

Submitted on 9 Mar 2017

Event based Identification and Prediction of Congestions in Manufacturing Plants

Clemens Schwenke[1], Thomas Wagner[1], Klaus Kabitzsch[1]

[1]Technische Universität Dresden.
{clemens.schwenke, thomas.wagner2, klaus.kabitzsch}@tu-dresden.de

Abstract. In modern manufacturing plants, most prominently in semiconductor manufacturing, complex interwoven automated material handling systems are used to transport lots between productive service stations. In busy areas, these systems often show transient congestion phenomena that have a negative impact on the throughput of the factory. Because of the systems complexity and the large amount of data originating from several thousand transports a day, these congestions are difficult to detect and to trace or even to predict. In this paper, the authors present an approach to detect congestions in such systems using an event-based model building and analysis approach. Once such effects have been identified, an algorithm is proposed to derive congestion prognosis rules which can then be used to implement effective congestion prevention mechanisms.

Keywords: Semiconductor AMHS, Model building, Event analysis, congestion prevention

1 Introduction

In modern semiconductor industry, more and more highly integrated customized wafer products have to be produced in shorter and shorter periods of time.

Consequently, a large amount of production steps for a big variety of different products is carried out on the one hand. On the other hand, the production equipment is used flexibly, so that the transport system in a highly automated factory has to be adjusted frequently to new routes for wafer transports between stations. In general, these stations are connected by automated material handling systems (AMHS) which are complex interwoven networks of transport elements such as conveyor belts, rotary tables and other handling devices.

Besides adjusting the transport system to new demands, the operating engineers of automated material handling systems face on main problem. AMHS often show congestion phenomena, which reduce the throughput in a wafer fabrication facility (fab). Most of the times, congestions result in queues of work pieces waiting for other work pieces because stations are temporarily overloaded or switched off (down times).

adfa, p. 1, 2011.

Modern material handling systems provide features to detect and adjust to undesirable situations automatically and resolve congestions. For this feature intelligent material flow routing rules have to be implemented as well as rules for altering the feed of new work pieces entering the system.

But in real fabs, practitioners ask themselves the following question. Which rules have to implemented, and more important, how to determine these rules systematically?

In order to analyze the overall performance of transport systems, event data of material passing certain waypoints of the system has been collected in log files.. But the task of analyzing transient congestions and extracting conclusions often is still a sisyphean undertaking because of three reasons. First, this task is still carried out mostly manually by visually inspecting log files. Second, sometimes the efforts of studying log files do not result in generally applicable rules. Third, sometimes congestions seem unexplainable.

In order to free the expert from this time consuming task, this paper introduces an approach to how event based congestion analysis and prediction can be executed automatically.

Consequently, the steps of an approach for semi automatic event data inspection are provided. These steps include collecting of relevant event data, building a state model of the transport process, identification of temporarily overloaded segments, backtracking to influencing segments and congestion analysis in order to extract rules for prediction of congestions.

As a result, rules for the prediction of congestion occurrence are derived. For validation, these rules are applied to new event data of the same AMHS, so that to predict congestions were predicted early enough for operators being able to take actions. For an exemplary use case, a set of trace data of wafer lots in an automated production line has been used for proving the approach. All steps of this workflow have been implemented in a software framework and tested against real fab data.

This paper is structured as follows. In Section 2 related work is considered. The approach for data inspection is described in Section 3. Section 3 includes an exemplary validation. Conclusions are drawn and an outlook is given Section 4.

2 Related Work

The authors investigated several possibilities to identify and analyze congestions in a given transport system. Consequently, some relevant approaches are discussed and the disadvantages of those, that initially seem to be most obvious and useful, are portrayed. The first thought when transport systems shall be examined is queuing systems. But the pure queuing theory could not easily be applied for the authors real world use case, because arrival rates, service durations and sometimes even capacities of the system's elements change constantly. Consequently, the second thought is time series. But the application of pure classic time series analysis was not feasible, because models could not explain the observed phenomena exact enough, nonsense correlations were found, or the calculation costs were too high. Third, the authors considered state model building for examining event sequences.

Finally, the authors combined findings from several specific fields. Therefore, the consideration of related work covers work in state model building, queuing theory and time series.

2.1 State model building

Automatic state model building requires event logs as input data and provides highly aggregated information about state changes in a system. The prerequisite is that recorded events can be understood as notifications of state changes of entities. Briefly, the main essence of state model building is the extraction of a graph out of traces of events.

In the case of material flow systems, events are recorded when loads enter workstations where they are processed, or when they enter conveyor segments where they are transported to a succeeding workstation. In the extracted graph, nodes represent states of loads being in a certain station or transport segment, and edges represent the transfer of a load into another station or transport segment. For this kind of event data, an event always indicates that a load entered a work station or transport segment. The use of discrete state models describing a system's (or device's) behavior as a sequence of possible steps was studied successfully before [1].

On the one hand, state models are useful to monitor or identify business processes [2]. Van der Aalst et al. [3] used state model building as a method for analyzing business processes, where events are generated when certain work steps begin and end. In so-called process mining, models of business processes shall be recovered or checked. Additionally, the relevance of business steps can be evaluated and performance indicators are calculated based on event logs. The main problems are to recover adequate models and to identify relevant process steps, since the log data of business processes, involving humans and external events, oftentimes contain non-deterministic portions. The resulting models then have to be mostly analyzed manually, sometimes including a few automatically calculated performance parameters if applicable.

On the other hand, state model building can be used for the examination of event logs of machines or transport systems, for example in semiconductor industry or in logistics applications. Compared to extracted models of business processes, the extracted models of logistic and manufacturing applications are more deterministic but contain many more states, so that sophisticated tailored analysis approaches are necessary to detect and explain unwanted phenomena, such as extremely varying delays [4] or changing reject occurrence [5].

In order to enrich a pure state model with more information, Vasyutynskyy suggested the combination of state model building with calculation of performance indicators, such as overall throughput times, holding times and inter arrival times on states. Consequently the result is called an extended state model [6].

State models can be used as the basis for detailed analysis of congestions if they manifest as tailbacks of loads waiting for preceding loads [7]. In this work, an approach to automatically carry out transient tailback recognition and cause identification was introduced. In order to identify origins and causes of these observed tailbacks, historic event log data of loads passing certain waypoints were inspected. The approach is based on analysis of holding times and capacities of transport segments. As a result, complete lists of tailbacks and affected segments are provided. Plus, for each tailback an initial cause event is determined. But this tailback analysis approach does not relate the occurrence of tailbacks to the constantly altering arrival rates of new loads entering the system. Therefore it was necessary to investigate different approaches to enable a successful prognosis of congestions.

2.2 Queuing Theory

Queuing theory is a tool for estimating performance indicators in networks of waiting lines and service stations. The service stations take a certain amount of time, e.g., for processing one work piece. The work pieces, or loads, travel through the system and wait in line in front of the service stations, thus forming queues.

The main application is the design of queuing systems, [8], [9]. At the design time important questions are: What is the average queue length, how long are average waiting times Wq, and how many service stations are needed? For answering the question of the average waiting time in the queue Wq, Formula (1) can be used [10].

$$W_q = W - \bar{s} = \frac{\lambda}{\mu(\mu - \lambda)} \tag{1}$$

The arguments for this Formula are the complete waiting time W that was spent in a system of queue and service station. The time W includes the average service duration . Alternatively the time Wq can be calculated using the arrival rate λ and service rate μ.

The basic assumption in queuing theory is stable arrival-and service rates. In contrast, these rates change frequently in the investigated real world systems, e.g. depending on product mix and order situation. As a result, Formula (1) for estimating Wq, was not directly applicable. The average time loads spend in a conveyor segment is called in the following.

2.3 Time series analysis

Classic time series analysis provides many disciplines [11]. For this work the most important ones are the following. The first one is time series analysis in the time domain, where oftentimes trends and seasons are extracted by developing linear models until the residues cannot be minimized anymore and are similar to stochastic white noise. This approach is used in, e.g., economics, biology and agriculture [12]. Sometimes this approach is also used in physics or engineering but only as a last resort if model building using known facts did not provide useful results [13]. For example, time series analysis is used in the field of predictive maintenance to model trends, seasons and noise of deterioration indicators [14]. But these approaches try to smooth of outliers instead of explaining them.

The second discipline often is applied for modeling the remaining residues, after trend and season are extracted, by estimating auto regressive moving average (ARIMA) models. ARIMA models often are used to model processes in economics, especially in financial industry trying to predict effects in the stock market [15]. This is done by assuming the stochastic nature of the unexplainable processes [16], [17]. Therefore, the main ingredients of these models are two parts, the auto regressive (AR) part and the moving average (MA) part. The AR part tries to model the time series by explaining the current value mainly by the previous value. The MA part models a white noise so, that in conjunction with the AR part the given time series can be approximated. Unexplainable peaks in general are considered outliers and are smoothed [18].

In contrast, the authors needed to explain the outliers, instead of smoothing them. As a result, the above mentioned time series analysis approaches were not applicable. One reason for this is that in reality peaks are not always stochastic and do not solely depend on the previous value.

2.4 Summary

When first confronted, the authors tested the following approach. First, the inter arrival times and holding times on conveyor segment in front of stations or rotary tables were examined for aggregated periods of time. With Formula (1) of the queuing theory, the waiting times W at stations were estimated at these time periods, but they did not match the actually observed holding times.

By a different approach, the authors aimed to produce forecast models in order to estimate autoregressive moving average (ARMA) models for inter arrival times and holding times. The predictions of these models were used to estimate the current waiting time W applying Formula (1) to the forecasted arrival rates.

Unfortunately, the quality of these models was not good enough for reliable predictions, because of one important fact. Forecast models tend to smooth peaks, because most of the times they are considered outliers. But in contrast, in the use case of investigating transport system data, the peaks of holding times are the sought after and to be explained congestions. Consequently, generic naive time series analysis was not constructive.

As examined by the authors, the key to understanding the establishing and resolving of congestions is the combination of system knowledge with time series analysis. Congestions can travel through the system like waves, superimpose and thus, cause significantly varying waiting times on certain conveyor segments in front of service stations. Therefore, there is a true relation between only certain arrival rates, service rates and waiting times. As a result, the authors integrated a step in the overall approach that selects only the relevant time series before they are investigated further.

3 Analysis Approach

The suggested analysis approach consists of a workflow of five general steps, see Figure 1. First event data of the AMHS has to be collected.

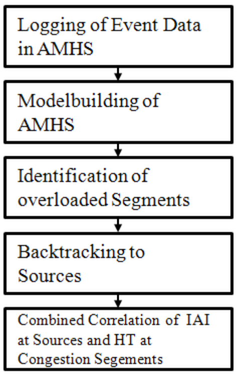
Logging of Event Data in AMHS
Modelbuilding of AMHS
Identification of overloaded Segments
Backtracking to Sources
Combined Correlation of IAI at Sources and HT at Congestion Segements

Fig. 1. Workflow of Analysis of Congestions

Second, a state model has to be extracted from of this event data. Third, overloaded segments of the transport system have to be identified. Fourth, the relevant source segments that feed loads into the system are identified by systematic backtracking.

Finally, the ultimate purpose of analyzing congestions by correlating them with arrival rates at source segments can be carried out. The four first steps are prerequisites for the last step. All five steps are described in detail in the following.

3.1 Logging of event data

The first step is the collection of event data in the factory's AMHS. That is, at each relevant conveyor belt or rotary table an event is logged. The event contains the essential information in the three fundamental attributes, timestamp, segment number and load number. Based on these elementary attributes, a graph of the transport system can be built in the second step.

3.2 State model building

In this step, a state model of the transport system is extracted from the logged event data. The authors published applications of this method before in [7], [19]. But for completeness, the algorithm of the method isbriefly described.

The algorithm extracts all relevant entities for building an extended state-transition model of a given log file of a given transport system. Consequently, the resulting model will consist of the following entities.

$$S = \{s_1; s_2; ... ; s_n\} \tag{2}$$

S is the finite not empty set of states, representing transport system elements, e.g., rotary tables or linear conveyor modules as well as storage elements (stockers) or production equipment (work stations).

$$L = \{l_1; l_2; ... ; l_m\} \tag{3}$$

L is the finite set of loads, representing the moved entities, e.g., wafer carrier.

$$T \subseteq S \text{ X } S \tag{4}$$

T is the finite not empty set of transitions, representing interconnections between the single elements. T is a subset of all ordered pairs of states. One element of T is a binary relation over S. For example, $(s_1; s_2) \in T$ with $s_1; s_2 \in S$; represents a transition from State s_1 to s_2. For instance, a rotary table can be used as a crossing, unification or split of transport streams. Therefore it is connected to several other elements and can exhibit multiple transitions.

The event log is an ordered sequence of events E as follows.

$$E = \{(\tau_1; s_1; l_1); ... ; (\tau_N; s_N; l_N)\} \tag{5}$$

One event $e = (\tau; l; s) \in E$ is defined as a triple consisting of timestamp $\tau \in Z$, state $s \in S$ and load $l \in L$. Z is the set of timestamps τ, so that $Z = \{\tau_1;...; \tau_N\}$. The above mentioned entities S, T and L can be systematically extracted from this ordered sequence of events as shown in Figure 2.

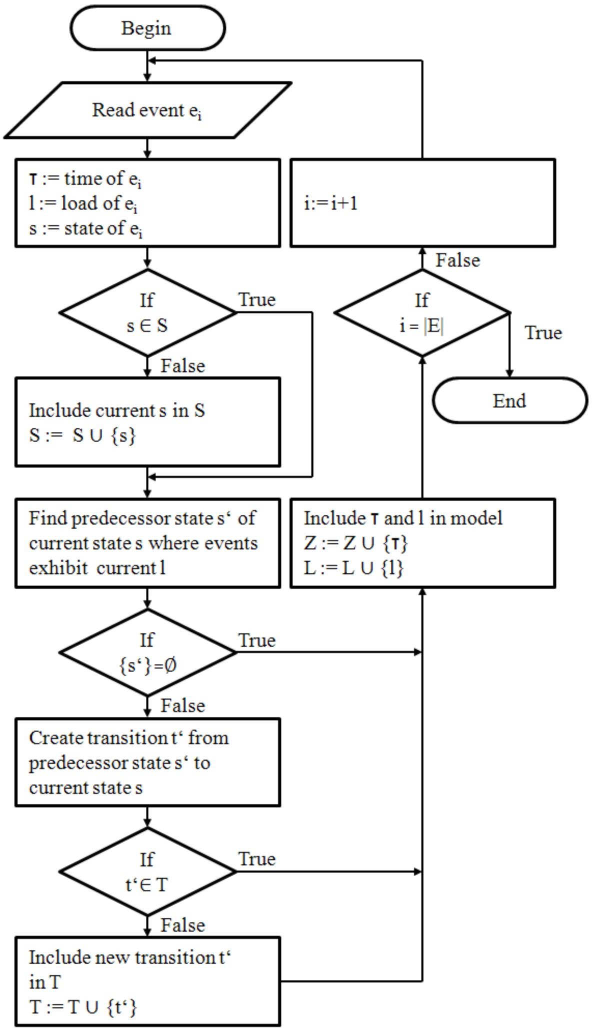

Fig. 2. State Model Building

The model building algorithm is a loop that processes ach event separately in one individual loop cycle. This loop consists of steps for extracting elements from events as well as for finding or creating model entities, so that they can be included into the model. Conditional decisions allow for breaking out of the loop if not all steps have to be carried out, because current entities are already part of the model.

3.3 Identification of overloaded segments

After the state model is extracted, the third step of the overall workflow can be executed. Overloaded segments are results of congestions. In the state-transition model these segments are states. The suspect states are identified by finding states that sometimes exhibit unusual long holding times ω.

Longer holding times are an effect of previous loads holding up succeeding loads and therefore affect the average holding times of loads on certain states.

For each state one corresponding average holding time $\omega(s_i)$ can be calculated. The states that exhibit unusual long average holding times compared to the average holding time over all states, $\omega(s_i) > \overline{\omega}$, are suspects to be affected by at least transient congestion effects. This comparison has to be executed for many fractions of time.

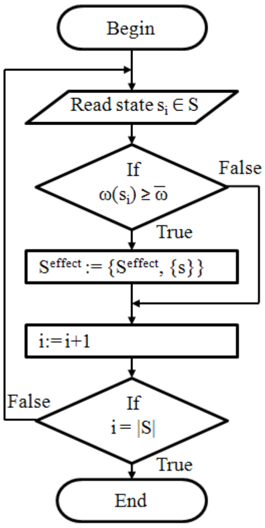
Begin
Read state $s_i \in S$
If
$\omega(s_i) \geq \overline{\omega}$
False
True
$S^{effect} := \{S^{effect}, \{s\}\}$
i:=i+1
False
If
i = |S|
True
End

Fig. 3. Identification of States that temporarily exhibit congestions

As a result, a set of congestion states $S_{effect} \subseteq S$ is found.

3.4 Backtracking to influencing segments

In order to find states that influence the holding time of congestion states, a backtracking is carried out. This is necessary to compare the time series of only those states that actually can have an influence and not others that are unlikely to have an impact.

This backtracking is carried out for each congestion state. The relevant influencing states are called feeding states. In that context, a feeding state is the closest preceding state that either exhibits more than one outgoing transition $d_{out}(s) \geq 2$ or that is a load source of the system, e.g. a production equipment input. Other states that represent linear conveyor segments and are closer do not have to be considered because the arrival rates do not differ from the feeding state.

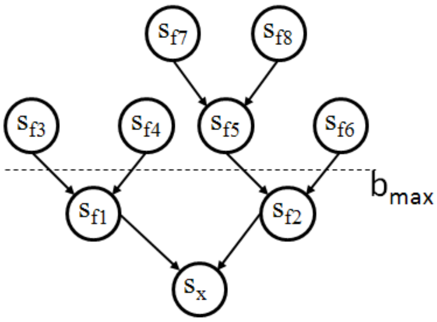

Fig. 4. Tree of influencing neighbor feeding states. Result of backtracking algorithm.

On each identified feeding state, recursively the same backtracking to previous feeding states is carried out. This recursion terminates when a number of maximum backtracking depth b_{max} is reached or if no more preceding states can be found in the state-transition model.

The result of this algorithm is a tree of feeding states for each congestion state, for example see Figure 4. The dashed line marks the maximum backtracking depth selected by the user. In this case, the time series of two feeding states sf1 and sf2 have to be considered in the congestion analysis step described in the next Subsection. Increased backtracking depths can result in longer forecast lead times for congestions but also cause higher calculation costs since more states have to be considered.

3.5 Congestion analysis

Once the above mentioned prerequisites are available, the actual congestion analysis can be started. The approach presented focuses on the diagnosis and prediction of tailback events caused by the dynamic interactions of different transport system elements or areas. Other possible causes of tailbacks, like random failures of single transport elements, are much less related to the system behavior which is observable using the event logs described in Section 3.1 and are therefore not considered. However, the prediction of such tailbacks could be tackled using semantic information about the transport systems hardware, e.g. mean time between failure considerations.

Here, the progression of the inter arrival times (IAT) of the identified feeding states are considered in order to identify conditions that provoked the anomalies on the congestion states. This approach allows to draw inferences from temporary different workload situations on different load sources about the manifestation of transient tailbacks, e.g. due to temporary load concentration or mutual obstruction.

Depending on the situation, not every feeding state identified in Section 3.4 has an influence to the appearance of tailbacks on the congestion states. To select the relevant subset of feeding states, several methods of selection can be considered. One approach could be to weigh the IATs of the different feeding state and ignore the ones that transport only a seemingly irrelevant amount of lots.

However, the simplified example depicted in Figure 5 suggests that this is a misleading approach.

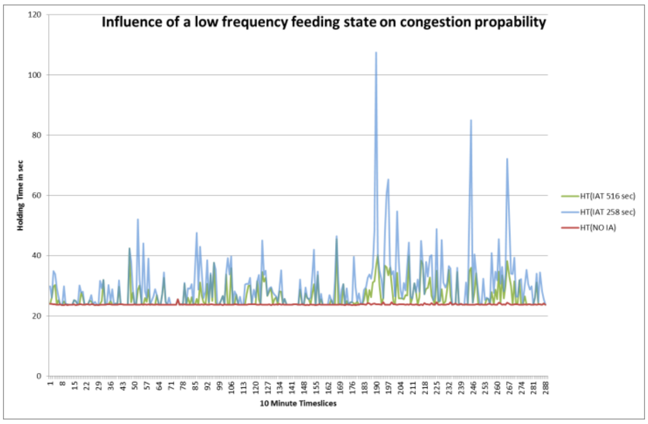

Fig. 5. Influence of a low frequency feeding state on congestion probability

In Figure 5, a congestion state A (see Figure 6) is shown which receives its loads at a rate of approximately three loads per minute from a major source B. If no other sources participate, no congestions appear at this state A as indicated by the red line. However, there exists another feeding state C which considerably influences the holding times on state A. Although state C only contributes to the traffic with around one load in nine minutes (green line) to around one load in 4 minutes (blue line), it significantly increases the holding times of A and therefore even causes congestions (noticeable peaks in the blue line).

In the presented case, this is caused by a dead-lock prevention mechanism implemented in the transport systems controllers that block all traffic from C down to D once a load enters the critical area shown as a red cuboid in Figure 6.

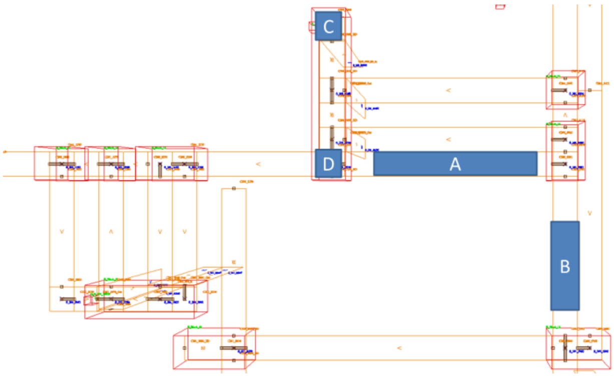

Fig. 6. Excerpt of the example system showing a congestion state A and the feeding states B and C

Therefore, it is necessary to measure the real influence of a feeding state's IAT on the HT of a congestion state regardless of its arrival rate. To achieve this, a correlation approach is used as a first step. For this purpose, the time series of holding time of the congestion state is correlated with the time series of the inter arrival times of all identified feeding states.

Depending on their distance, it takes a certain amount of time until the inter arrival times of the feeding states affect the holding times of the congestion state. To take these delays into account, each IAT time series is cross-correlated with the HT time series, using a default maximum lag as defined below where N is the number of observations in the time series.

$$maxlag = 10 * log_{10}\left(\frac{N}{2}\right) \tag{6}$$

As a second step, in the resulting correlation values r_l for each lag l, it is then sought for the maximum negative correlation. That is, the maximum suggested impact of a decreasing IAT (increasing arrival rate) of the feeding states on increased HTs of the congestion state. This correlation value r_l is then checked for significance by comparing it with the values of the approximated 99% confidence interval c ($\propto$=0.01):

$$c = \pm\frac{2.58}{\sqrt{N}} \tag{7}$$

Third, once the set of significant feeding states S^sig has been found and the corresponding lags $lag_{crit}(s^{sig})$ yielding the maximum negative correlation for each state noted, the critical inter-arrival rates must be identified. These are the ones that cause overload situations on the congestion state, if they appear in combination. For this purpose, a set V for each feeding state $s^{sig} \in S^{sig}$ is constructed as follows.

$$V = \left\{\bigvee_{n\in(N-l)} v_n\left(IAT(s^{sig})_n; HT(s^{effect})_{n+l}\right)\right\} \tag{8}$$

With $\mathrm{l} = \mathrm{lag_{crit}}(\mathrm{s^{sig}})$. In summary, this set V contains a mapping from the inter-arrival times of one feeding state to the corresponding holding times of the congestion state, shifted by l to compensate for the time delay between cause and effect as mentioned above. Afterwards, V is sorted by its IAT values in descending order, i.e., from the least to the most frequent lot appearance. Next, the HT components of V are scanned along the falling IAT values. Once the holding time on the congestion state first reaches or exceeds the critical value $\mathrm{HT_{crit}} = \mathrm{HT}\left(\mathrm{s^{effect}}\right) \geq \overline{\omega}\left(\mathrm{s^{effect}}\right)$ as described in Section 3.3, a previous $\mathrm{IAT}(\mathrm{s^{sig}})_{\mathrm{crit-k}}$ is used to define a rule indicating the danger of congestion. This rule sets a Boolean value $\mathrm{d}\left(\mathrm{s^{effect}}\right)$ depending on whether this value $\mathrm{IAT}(\mathrm{s^{sig}})_{\mathrm{crit-k}}$ is undershot.

The parameter *k* can be used to manipulate the lead time and sensitivity of the congestion prognosis by choosing more conservative, i.e. larger, values of $\mathrm{IAT}(\mathrm{s^{sig}})$, so that the warning signal $\mathrm{d}\left(\mathrm{s^{effect}}\right)$ is set earlier.

This procedure is repeated for every $s^{sig} \in S^{sig}$ and the resulting rules are combined to one single rule, suggesting a high congestion probability once all of the conditions are met.

The construction of these rules will now be demonstrated by the example shown in Figure 6. In this example, the state A's relevant HT_{crit} was found at 60 seconds.

In Figure 7 the ordered set V is shown for the congestion state A and the feeding state B.

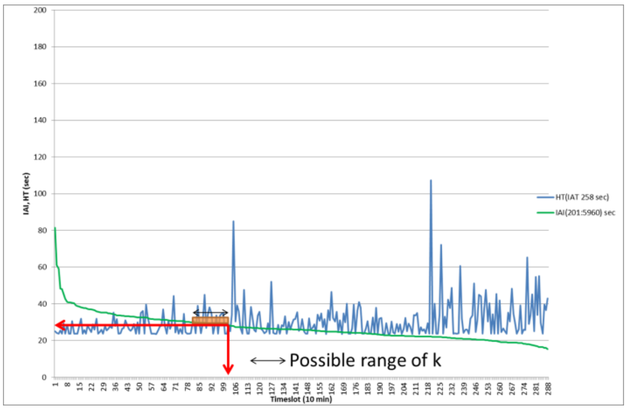

Fig. 7. Critical IAT for feeding state B

For this state, the congestion effects began to manifest once the IAT of state B was less than or equal to 27 seconds. Using a parameter value of 3 for the lead time parameter k, i.e., the next larger $IAT(s^{sig})$ value is used which is 28 seconds. In this example the congestion indication rule can be defined as follows.

$$d(s^{effect}) = IAT(B) \leq 28 \tag{9}$$

A second rule is derived from the holding- and inter-arrival times of the states A and C as shown in Figure 8.

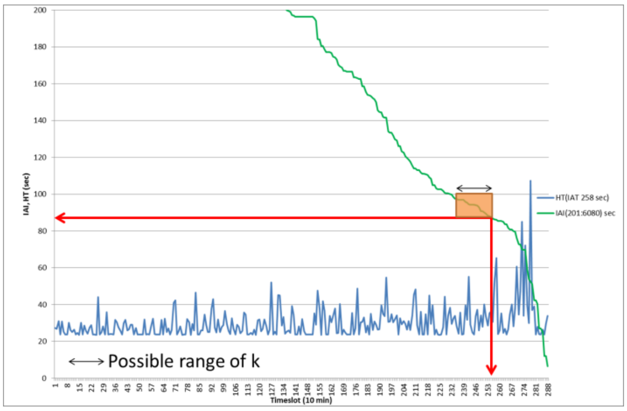

Fig. 8. Critical IAT for feeding state C

Here, congestions on state A appeared once the inter-arrival times of state C was lower than or equal to 86 seconds. Using again a k of 1, the corresponding congestion indication rule therefore is as follows.

$$d(s^{effect}) = IAT(C) \leq 88 \tag{10}$$

As a last step, the resulting rules must be combined to reflect the mentioned interrelation of the IATs between the corresponding states. A combined rule would be noted as follows.

$$0d(s^{effect}) = (IAT(B) \leq 28) \wedge \ (IAT(C) \leq 88) \tag{11}$$

4 Conclusion and Outlook

The presented approach has been implemented into a comprehensive analysis framework. User input is only required to define the input parameters k and the maximum backtracking depth b_{max} to influence the prognosis lead time. Subsequently, the warning rules are derived fully automatically and can afterwards be evaluated against the current transport system behavior at runtime.

The derived warning rules for congestion prognosis will serve as a basis for dynamic routing approaches within the transport system controllers. If their conditions are met, the controllers will be alarmed about possible future congestion situations. As a possible countermeasure, they can reroute part of the incoming traffic flow across different system parts, thus gaining a consistent lot flow while sacrificing only a small amount of transport speed for a few lots.

In the use cases investigated, the derived rules predicted the observed congestions accurately enough to allow for effective prevention measures in most cases. However, the approach also exhibited a few limitations that have to be considered regarding the choice of the parameter values k and b_{max}.

If the maximum backtracking depth is set too large, then too many feeding states have to be considered, eventually causing interferences between the growing variety of load situations. That means, several groups of feeding states may cause overload situations on a single congestion states independently. Using just the cross-correlation approach, this can neither be reliably distinguished nor can it be expressed using just AND conjunctions of the warning rules.

In addition, the lead time parameter k must be chosen carefully, since small values may reduce the prognosis horizon too much. On the other hand, values too large may provoke a lot of false-positive congestion warnings.

Therefore, future work will focus on defining metrics to aid the system experts choosing the right parameter values. In addition, the authors will investigate a wider set of influencing variables to determine their suitability for making the predictions more accurate.

5 References

1. Kemper, P., C. Tepper. “Trace based analysis of process interaction models”. In Proceedings of the 2005 Winter Simulation Conference (WSC), 427–436. 2005.
2. Agrawal, R., D. Gunopulos, F. Leymann. “Mining process models from workflow logs”. In Proceedings of the Advances in Database Technology (EDBT), 6th International Con-

ference on Extending Database Technology, Valencia, Spain. Lecture Notes in Computer Science, vol. 1377, 469–483. Berlin: Springer-Verlag. 1998.

3. van der Aalst, W. M. P, H. A. Reijers, A. J. M. M. Weijters, B. F. van Dongen, A. K. Alves de Medeiros, M. Song, H. M. W. Verbeek. “Business process mining: An industrial application”. Information Systems, vol. 32, no. 5, 713–732. 2007.
4. Gellrich, A., T. Wagner, V. Vasyutynskyy, K. Kabitzsch. “Modeling of Transport Times in Partly Observable Factory Logistic Systems based on Event Logs”. In Proceedings of the 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011), Toulouse, France. 2011.
5. Shanthikumar, J. G., S. Ding, and M. T. Zhang. “Queueing theory for semiconductor manufacturing systems: a survey and open problems”. IEEE Transactions on Automation Science and Engineering, vol. 4, no. 4, 513–521. 2007.
6. Vasyutynskyy, V., A. Gellrich, K. Kabitzsch, and D. Wustmann. “Analysis of Internal Logistic Systems Based on Event Logs”. In Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010), Bilbao, Spain. 2010.
7. Schwenke, C., T. Wagner, A. Gellrich, and K. Kabitzsch. „Event-based recognition and source identification of transient tailbacks in manufacturing plants”. In Proceedings of the 2012 Winter Simulation Conference (WSC), (pp. 1-12). IEEE. 2012.
8. Beranek, L.. “A Method of Predicting Queuing at Library Online PCs”. Australian Academic and Research Libraries, Vol. 37 Issue 4, p260-272. 2006.
9. Horling, B., and V. Lesser. “Using Queuing Theory to Predict Organizational Metrics”. Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, ACM, pp. 1098-1100. 2006.
10. Gross, D., C. M. Harris. “Fundamentals of queuing theory”. New York: Wiley, 1998.
11. Hamilton, J. D. “Time series analysis”. Vol. 2. Princeton: Princeton university press, 1994.
12. Mead, R., R. N. Curnow, A. M. Hasted, and R.M. Curnow, “Statistical Methods in Agriculture and Experimental Biology”, 2002.
13. Palit, A., and K. Ajoy, “Computational intelligence in time series forecasting: Theory & engineering applications”, 2005.
14. Krause, J., and K. Kabitzsch, “A generic Approach for Reliability Predictions considering non-uniformly Deterioration Behaviour”, Journal of Physics: Conference Series, Volume 364, Issue 1, 2012.
15. Wang, J.-H., and J.-Y. Leu, “Stock market trend prediction using ARIMA-based neural networks”, IEEE International Conference on Neural Networks, Volume 4, Pages: 2160 – 2165, 1996.
16. Bollerslev, Tim, H. O. Mikkelsen, “Modeling and pricing long memory in stock market volatility”, Tim Bollerslev, Hans Ole Mikkelsen, Journal of Econometrics, vol 73, issue 1, 1996, pp151-184
17. Nelson, Daniel B. The time series behavior of stock market volatility and returns. Diss. Massachusetts Institute of Technology, 1988.
18. Breen, William, Lawrence R. Glosten, and Ravi Jagannathan. "Economic significance of predictable variations in stock index returns." The Journal of Finance 44.5 (1989): 1177-1189.
19. Wagner, T., C. Schwenke, K. Kabitzsch. Modeling and wafer defect analysis in semiconductor automated material handling systems," In Proceedings of the 2012 Winter Simulation Conference (WSC), pp.1-12. 2012