Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

AUTOMATED MODEL DEVELOPMENT AND PARAMETRIZATION OF MATERIAL FLOW SIMULATIONS

Michael Milde
Gunther Reinhart

Institute for Machine Tools and Industrial Management
Department of Mechanical Engineering
Technical University of Munich
Boltzmannstraße 15
Garching b. München, 85748, GERMANY

ABSTRACT

Material flow simulations are a powerful tool for planning and improving complex production systems. In practice, however, the potential of simulations cannot be exploited. Long development times and high efforts complicate a successful realization of simulation projects. In this paper we present a new approach to automate time-consuming steps in simulation projects. Based on general tracking and tracing data, our approach reduces the efforts in the area of data acquisition and automates the steps of model development, model parameterization and model implementation. The approach automatically identifies a material flow model, classifies and parametrizes the identified material flow elements and extracts basic control policies. A first prototypical implementation confirms the validity of the approach. The results show that the simulation models can be successfully created automatically with low manual effort.

1 INTRODUCTION

In a more and more complex world, the ability to adapt quickly to new circumstances will be a key competitive advantages (ElMaraghy 2008). This requires manufacturing companies to make their production flexible and reconfigurable (ElMaraghy 2008). On the other hand the complexity of production systems rises, which makes it difficult to adapt these systems. In recent years, discrete event simulation (DES), referred to as simulation in the following, has proven to be a promising approach for leveraging these competitive advantages (Skoogh et al. 2012). Simulation allows to investigate the impact of changes in production quickly without interrupting the real production processes (Fishman 2001). Further, it increases the quality of planning by allowing dynamic and stochastic aspects to be taken into account (VDI 2016b).

Although being a promising tool, many companies cannot fully exploit the potential of simulations (Barlas and Heavey 2016). In practice, some shortcomings prevent the successful implementation of simulation projects. Long development times for high-quality models are in conflict with the desire for fast results (Skoogh et al. 2012). This leads to models with a lower level of detail and validity, especially in modeling stochastic and dynamic behavior of the system under study (Barlas and Heavey 2016). Further, the effort to keep simulation models up to date, leads to a low reusability of the models (Skoogh et al. 2012).

The aim of this paper is to tackle the described challenges and deficits in simulation development for existing production systems by presenting an approach for automated simulation model development and parametrization. The approach uses the data generated due to the ongoing digitization in manufacturing and the new possibilities to extract knowledge from this data through data mining methods.

The paper is structured as follows. In Section 2 we describe the common steps in simulation projects and their challenges resulting in the mentioned deficits. In Section 3 we review previous literature related to automatic model generation and parametrization. In Section 4 we present our approach, the prototypical implementation and validation of which is discussed in Section 5. In Section 6 we critically discuss our approach and conclude our findings in Section 7.

2 PROCEDURE IN SIMULATION PROJECTS AND THEIR CHALLENGES

To give simulation projects a general structure, several project management frameworks proposing the key steps in simulation projects have been developed. Figure 1 shows the procedure model published by VDI (2016b) and the relative effort each individual project phase contributes according to a study with simulation practitioners (Mayer and Mieschner 2017). The phases of data management and model development are the most time consuming steps in simulation projects. Looking more deeply into these steps we can further identify the key challenges.

A major challenge in data management is the often inadequate availability of data. Input data for the simulation are not directly available and data sources must first be identified (Skoogh et al. 2012). Due to the different data sets required for simulations, such as processing times, machine arrangement, malfunction data, etc., these are usually not found in one but in several heterogeneous IT systems (Skoogh et al. 2012). In addition, most data sets do not have the correct form and structure for direct use in simulations and must therefore first be converted into the corresponding form (Barlas and Heavey 2016). Often raw data sets have to be analyzed manually to extract the needed parameters. For the modeling of dynamic and stochastic aspects, additionally a large amount of measurement data of the real system are required to ensure valid modeling (Skoogh et al. 2012).

The great challenge in the field of modeling lies in the high complexity of the production systems under investigation (Perera and Liyanage 2000). Due to increasing product individualization, the number of variants and thus also the complexity of production increases, be it due to higher control effort producing several different products in one system and a greater number and heterogeneity of machines and process steps. The simulation expert must understand this system complexity in order to be able to model the system. The expert often encounters distributed development teams during the information collection phase. As a result, the number of production experts to be questioned increases, which lengthens the system analysis step and thus project duration.

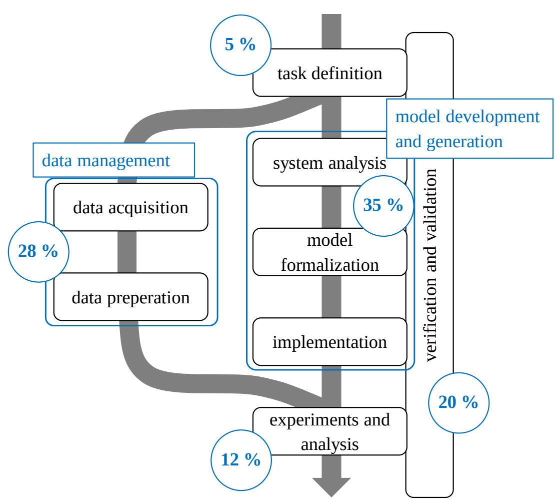

Figure 1: Procedure model for simulation projects according to VDI (2016b) with the relative effort of the individual project phases (Mayer and Mieschner 2017).

3 AUTOMATION OF SIMULATION PROJECTS: STATE OF THE ART

In literature, we find several approaches to reduce development times in simulation projects. These approaches automate different phases in simulation projects. We can distinguish between three areas in which automation is used. (1) covers model development and thus belong to the steps system analysis and model formalization. (2) refers to parametrization of the model or Input Data Management (IDM), which, according to Skoogh et al. (2012), includes all steps from raw data through their transformation to data that is usable in the simulation model. In Figure 1 this belongs to the steps data acquisition and data preparation. (3) corresponds to the implementation of the formal model, resulting in an executable simulation model. This step is also referred to as model generation (Bergmann and Straßburger 2015).

Aufenanger et al. (2010), Skoogh et al. (2012) and Barlas and Heavey (2016) propose approaches for the an automated IDM corresponding to approaches of area (2). The authors follow a similar scheme with their approaches. The authors develop methods to extract data from heterogeneous data sources, then transform this data, perform mathematical calculations, generate statistical representations of the data and finally link them with the corresponding parameters in the simulation model. All approaches are based on the fact that the simulation model already exists and only needs to be parameterized. The steps in the area of model building must therefore already have been carried out and are therefore not part of the approach. Each approach requires a manual initial configuration, but can then perform the IDM steps automatically. The strength of these approaches can be seen in updating the input parameters of the simulation, which is executed completely automatically.

Smith (2015) uses the data from GPS trackers mounted on parts as a basis for an integrated approach automating all phases from model development, parametrization, and model generation. Based on location and acceleration data, the author identifies process stations and logistic connections, extracts their properties and reconstructs the material flow in the production. A model generator implements an executable simulation model in a proprietary simulation software. Extraction of dynamical behavior like control policies is not reported by the author. As this results in not fully complete simulation models, the author suppose to add the missing aspects to the model manually.

Charpentier and Véjar (2014) follow a similar concept for the automated reconstruction of material flows and their transformation into an executable simulation model. The authors do not use GPS data but event data, which contain an exact identification of each part in production, their positions as well as a time stamp indicating the arrival time at the given position. An explicit naming of the data sources, from which this data originates, is not given by the authors. The authors' approach is able to detect changes in the material flow during manufacturing operations and update the simulation model online. Similar to the approach of Smith (2015), Charpentier and Véjar (2014) do not identify specific control policies, like part routing or priority rules.

The approach of Selke (2005) concentrates on the automated extraction of technical data and system load data on the basis of shop floor data collection. In addition, the author describes strategies for automated identification of control policies in the production systems, such as batch size and sequence design. However, the approach still requires a lot of manual effort in data acquisition and preparation. Further, the identification of control policies is not integrated in an overall automated process for model development.

In summary, it can be stated that a general and integrated approach for automated simulation model development and parameterization has not yet been developed. The existing approaches usually only focus on partial areas, such as model creation or continuous updating of model parameters. Only the approaches presented by Smith (2015) and Charpentier and Véjar (2014) describe an integrated method automating model development and IDM. However, these approaches require specialized input data and are not able to identify control policies. Therefore both do not fulfill the requirement of modeling dynamic aspects. This shortcomings can be solved by methods presented by Selke (2005), who focuses on the identification of control policies. His approach requires manual initial effort and therefore is not fully automated. In addition, the approaches do not sufficiently describe the data they use or place high requirements on its quality and structure. This reduces the broad applicability, so that practitioners can hardly profit from these approaches.

4 AN APPROACH FOR AUTOMATED SIMULATION MODEL DEVELOPMENT AND PARAMETRIZATION

The goal of our approach is a more efficient execution of material flow simulation projects for existing production systems. The developed approach automates the phases of model development and input data management. Figure 2 shows our general concept. In the first step, the simulation expert extracts tracking data and data on machine malfunctions and breakdowns from manufacturing data acquisition systems. The expert then transfers the data into a table structure defined by our approach. These steps are performed manually. Based on this data the algorithm reconstructs the material flow and classifies the identified material flow objects. This results in the material flow model of the system under investigation, which is subsequently parameterized in the next step. The resulting parameterized material flow model serves as a basis for the following identification of control policies. After completion of this step, we receive a fully described formal model of the production system, which is stored and exported in a general exchange format for simulation models. This file holds the input data for an automated model generator, which generates an executable simulation model. All steps from reading the raw data to model generation are performed automatically. In this paper, we focus on the first steps beginning with the identification of the required raw data until the extraction of the formal simulation model. In the following, we describe these steps more detailed.

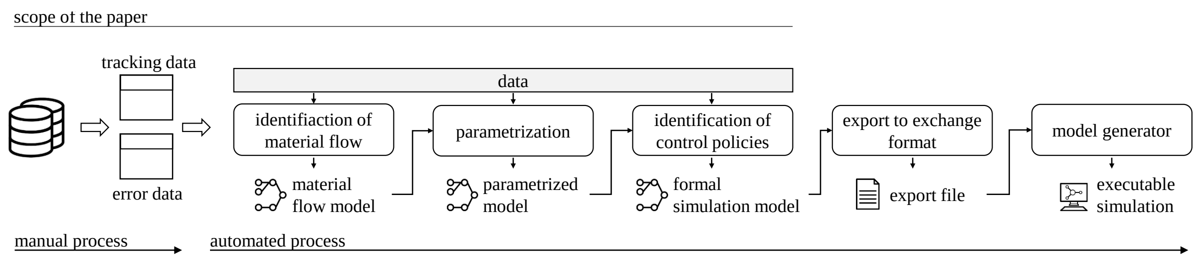

Figure 2: Graphical representation of process steps of our approach for automated simulation model development and parametrization.

4.1 Identification and Preparation of Primary Data

The aim of material flow simulations is the dynamic modeling of system states over time (VDI 2016b). These system states are on the one hand the positions of parts in the production system and on the other hand the conditions of the production resources. In order to be able to create the simulation automatically, we need information about the dynamics of the real production system. This information must be based on the real system states that occurred in the past production process, in order to accurately represent stochastic and dynamic aspects of the system. Tracking and tracing data are available for this purpose, which record information about components passing defined points in the material flow. According to VDI (2016a), Vasyutynskyy et al. (2010), and Hippenmeyer and Moosmann (2016) tracking and tracing systems generally record the following information:

- related or passed production or logistic resource
- timestamps of start and end of processing steps, events, etc.
- involved employee
- production order data like involved materials, unique identifiers, customers, etc.
- process parameters like malfunction, errors, breakdowns, quality data, energy consumption, etc.

In order to be independent of proprietary tracking and tracing applications, we define a general input data scheme for our approach, derived from the information listed above. We define two table structures, the tracking table, containing data about the positions of parts in the system, and the error table, containing data about malfunctions and breakdowns of production resources. The tracking table contains the following information:

- **Part identification (ID)**: Unique identifier for each processed part.
- **Material**: Information, which product or material the processed part is.
- **Station**: Identifier / name of station / production process the part is currently processed at.
- **Start timestamp**: Date and time, when part enters the station.
- **End timestamp**: Date and time, when part leaves the station.

Our approach requires the recording of these data for each production resource, like assembly stations, machining cells, etc. in the production system under investigation. Tracking data of logistic processes like buffers, conveyor belts, etc. are not required by our approach as these systems are often not equipped with sensors for tracking and tracing in practice (Bergmann 2013). Table 1 shows an exemplary tracking table.

Table 1: Example table for tracking data.

Part ID	Material	Station	Start timestamp	End timestamp
001	material1	milling01	2019-04-15 11:05:31	2019-04-15 11:08:01
001	material1	milling02	2019-04-15 11:08:50	2019-04-15 11:10:00
001	material1	drilling01	2019-04-15 11:11:10	2019-04-15 11:15:00
002	material4	milling01	2019-04-15 11:08:02	2019-04-15 11:10:32
...	...	...	...	...

The error table contains the following information:

- **Error identification (ID)**: Unique identifier for each error occurred.
- **Station**: Identifier or name of station or production process the error occurred at.
- **Error type**: Identifier or name of the error type. We distinguish between technical errors, caused by malfunctions during the production process, and organizational errors, caused by a blockage in the subsequent material flow.
- **Start timestamp**: Date and time, when malfunction begins.
- **End timestamp**: Date and time, when malfunction ends.

Table 2 shows an exemplary error table. After the manual filling of the tables by the user, our approach reads in the data, whereby only complete entries, i.e. with filled start and end timestamps, are used.

Table 2: Example table for error data.

Station	Malfunction type	start timestamp	end timestamp
milling01	technical	2019-04-15 10:13:21	2019-04-15 10:39:41
milling02	technical	2019-04-15 10:48:50	2019-04-15 10:50:00
drilling01	organizational	2019-04-15 11:01:10	2019-04-15 11:02:00
...	...	...	...

4.2 Identification of Material Flow

When identifying the material flow, the material flow elements of the production system under consideration are first extracted from the data and classified, and then the connections between the elements are identified.

For this purpose, the routes of the individual components through the system are determined first. For each part, described by the part ID, all material flow events in the tracking table are sorted chronologically. This results in a specific route for each part. The frequency of the various routes is then counted for all part and a transport matrix is created. Thus, all possible connections between production resources are available in this matrix.

Real production systems consist of a large number of different material flow elements, such as machining and assembly stations, but also logistics elements, such as conveyors, warehouses and buffers. In material flow simulations, these elements and their properties must be modeled accordingly. A complete identification of all possible elements is difficult to achieve with an approach for automated model development. However, in principle material flow elements can be divided into production and logistics elements in a simplified way. In our approach we therefore distinguish between the elements of a processing station, as basic production element and a buffer, as basic logistics element. All other elements can be derived from these elements via appropriate parameterization and applied control strategies. As an example, a conveyor line can be interpreted as a buffer with defined maximum capacity, first-in-first-out logic and fixed throughput time.

The automated classification of production elements takes place directly from the tracking table. As we set the requirement for the input data, that each processing step in the material flow is equipped with a data collection point for recording incoming and outgoing components, it follows that all resources contained in the tracking table are characterized as processing stations.

In the following, all connections between the processing stations are examined. These are derived from the transport matrix. The time differences at these connections are then examined by comparing the end timestamp of the processing station with the start timestamp of the following processing station. If the time difference for a connection is close to zero, a direct chaining can be assumed. A component is transferred directly from one station to the next. If the time difference is higher, the connection is characterized as a logistical element buffer.

4.3 Parameterization of Material Flow Elements

For the parameterization of the identified material flow objects, time-related parameters such as shift calendars and process times are considered on the one hand, and malfunction behavior on the other.

In order to automatically derive the shift calendar of the production system under investigation, all time stamps from the tracking table are sorted chronologically. Since no material flow movements occur during breaks or outside of shift times, these can be determined by means of larger gaps between the time stamps. The chronologically sorted tracking table is searched for gaps that exceed a minimum threshold. The minimum threshold must be estimated by the user in advance. The shift calendar can now be reproduced on the basis of the gaps found.

The basis for the derivation of process times is the retention time of components on material flow elements. For the correct parameterization of the material flow elements, the standard time is decisive. The standard time is the time required to carry out an operation at normal power and corresponds to the target time for the execution of operations (Dangelmaier 2001). In the tracking table, the retention time of components on material flow elements is available as the difference between the end and start timestamp of each operation. During retention time, standard times, time periods of technical and organizational malfunctions, and shift times can overlap. This case is illustrated exemplary in Figure 3. To derive the standard time correctly, error periods from the error table and shift breaks must be subtracted from the retention times.

The standard times are again subdivided into set-up times and execution times (Dangelmaier 2001). This differentiation should also be parameterized in the material flow elements of the simulation. For the differentiation, the standard times calculated for each material flow element are sorted chronologically with the corresponding material number of the part. Now the material number is compared with the material number of the previous part processed at this material flow element. If the material numbers differ, the standard time consists of the sum of setup time and execution time. If there is no difference between the material numbers, it can be assumed that the execution time corresponds to the standard time and that no setup times occur. Execution time and setup time can fluctuate in real operation and therefore show stochastic behavior. This makes it difficult to strictly separate both time components. In our approach, we therefore determine a combined matrix of distributions of set-up and execution times. Rows and columns of the matrix describe the respective material numbers. The matrix fields contain the distributions of the measured target times according to the preceding material number indicated in the row and the material number of the component under consideration indicated in the column.

By the combined matrix structure and the determination of distributions stochastic aspects can be modeled realistically on the one hand and on the other hand a realistic modeling of set-up times and execution times can be achieved.

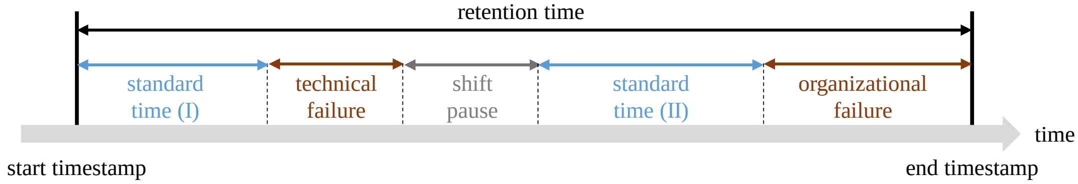

Figure 3: Different time periods of the part's retention time on a material flow element.

Machine breakdowns or malfunctions are modeled in simulations by specifying the availability V of the production resource in percent and the mean downtime, called mean time to repair $MTTR$. The parameters are calculated on the basis of the error table using the following formulas:

$$V = 1 - \frac{\sum_{i=1}^{L}(T_{i,\text{end}} - T_{i,\text{start}})}{T_E},$$

$$MTTR = \frac{\sum_{i=1}^{L}(T_{i,\text{end}} - T_{i,\text{start}})}{L},$$

where L is the number of errors, $T_{i,\text{end}}$ the end time of the ith error, $T_{i,\text{start}}$ the start time of the ith error, and T_E the recorded operating time of the resource. Only the technical malfunctions of the error table are included in the calculations, since organizational errors result from material flow relationships and the dynamics in the system.

4.4 Identification of Control Policies

In order to develop a valid simulation model, the control policies of the real production system must also be modeled. These include routing rules, which determine the path a part takes for several possible paths in the production system, and priority rules, which determine which component is processed next.

4.4.1 Routing Rules

Table 3 shows common routing rules that are predefined in DES software applications. If a rule is not feasible because a transfer of the part to the next element is disturbed due to an occupancy or failure of the subsequent material flow element, it can either behave 'blocking' or make a new feasible decision ('non-blocking'). The second column indicates whether the rule can be detected automatically (✓) or not (✘) by our approach. A '-' indicates that the rule cannot show the given behavior according to its definition. We describe the procedure to detect the individual rules automatically in the following.

Cyclic and cyclic sequence

Table 3: List of routing rules.

Routing rules	**Detection**	
	blocking	non blocking
Cyclic	✓	✓
Cyclic sequence	✓	✓
Start at successor 1	-	✓
Linear sequence	-	✓
Min. / max. contents	✓	✓
Random	-	✓
Percent	-	✓
Least / most recent demand	✘	✘
Min. / max. number in	✘	✘
Min. / max. processing time	✘	✘
Min. / max. setup time	✘	✘
Min. / max. relative occurrence	✘	✘
Part attribute	✘	✘
Defined successor	✓	-

In this rule, parts are distributed to the following elements according to a predefined pattern. Cyclic is the special case where the part is moved to all subsequent elements one after the other. In order to recognize this rule automatically, all part transfers to directly following material flow elements are determined and sorted chronologically. In the resulting list, Sequential Pattern Mining (SPM) is used to search for recurring patterns. We use the SPM method according to Yang et al. (2003).

During pattern recognition, it is possible that recurring patterns are found, but these do not correspond to a cyclic sequence or have arisen randomly. Therefore, the identified patterns are checked for their correctness using the chronological list of part transfers. If the pattern is deviated from, there are two possibilities. (1) The pattern is faulty and there is no cyclic sequence for this pattern. (2) The continuation of the pattern is not possible because the subsequent element is occupied or disturbed and the part is moved to the next possible successor. This can be determined from the input data. In this case a non-blocking behavior of the cyclic sequence is detected.

Start at successor 1 and linear sequence

With this rule, parts are forwarded to the next element according to a predefined priority list. In order to automatically detect this rule, the system first determines which subsequent element was occupied most frequently. This results in a prioritization list descending from the highest occupation. Afterwards, the prioritization is checked on the basis of the sequence of chronologically sorted part transfers from the current material flow element to its successors. If the part is not transfered to the prioritized successor, it is checked whether the successor is occupied or disturbed. If this is the case, the algorithm checks whether the component has been transfered to the next prioritized successor and so on. If the sequence is violated, a linear sequence can be excluded.

Min. / max. contents

With this rule, the component is transfered to the successor with the fewest / most parts already on it. In order to automatically recognize this rule, the system checks for each part transfer whether the successor had the smallest / largest contents. If this procedure is deviated from, the non-blocking behavior must be investigated. If the actually intended successor is completely occupied or disturbed, the algorithm checks whether it has been transfered to the successor with the next lower / higher contents. This is checked for all further successors. If the rule is also violated here, the rule 'min. / max. contents' can be excluded.

Random and percent
The rule 'Percentage' transfers parts to the successors after a defined probability distribution. 'Random' is the special case where every successor is given the same probability. This rule is only recognized by our approach if no other rule applies. For automatic recognition of the rule, the transfers to each individual successor are counted and divided by the total number of transfers.

Defined successor
With this rule, a fixed subsequent material flow element is assigned to each part type, defined via the material attribute from the tracking table. The rule can be derived directly from the transport matrices of the individual part types. This rule is a special case of the general rule part attribute.

Non detectable rules
The other rules listed in table 3 are not automatically recognized in our approach. The rules 'min. / max. processing time' and 'min. / max. setup time' are not detected, because an exact breakdown of setup and process times as described in section 4.3 is not possible with the defined input data. All other non detectable rules refer to additional attributes of parts. In the basic concept of our approach, these are not listed in the input data. To recognize routing rules based on attribute values, the corresponding attributes would have to be included in the input data.

4.4.2 Priority Rules

Priority rules are used for case-specific sequence sorting of parts. They are carried out on material flow elements with a part capacity greater than 1. According to Lödding (2016), they aim to increase the service level, delivery reliability or system performance. Here, many decisions are made about priorities on the basis of production order data. For example, the rules 'earliest plan start date' and 'earliest plan finish date' require planning data of production orders. Due to the minimal requirements to the input data, such priority rules can not be recognized in our approach so far. We therefore concentrate on the automated recognition of the strategies first-in-first-out (FIFO), last-in-first-out (LIFO), and setup time optimization.

FIFO
For automated recognition of FIFO, all incoming and outgoing timestamps with part IDs are sorted chronologically for each material flow element. If the incoming order of the part IDs corresponds exactly to the outgoing order, then the corresponding material flow element follows FIFO.

LIFO
For automated recognition of LIFO, all incoming and outgoing time stamps with part IDs are sorted chronologically for each material flow element. If the component ID of the outgoing part always corresponds to the ID of the last incoming part, then the corresponding material flow element follows LIFO.

Setup time optimization
This strategy tries to arrange as many parts of the same setup class as possible in order to minimize the number of setup processes. In order to automatically recognize this strategy, it is necessary to know the previous outgoing part for each outgoing part. For each material flow element, therefore, the first part that leaves the element is recorded first. All further outputs are examined chronologically. The setup time optimization strategy is only available if one of the following conditions is fulfilled: (1) The leaving part belongs to the same setup class as the part that left the material flow element before. (2) At the time of exit of the part, there are no other parts of the same setup class on the material flow element.

5 PROTOTYPICAL IMPLEMENTATION AND VALIDATION

A first prototypical implementation of the approach took place in Matlab. In the first step the Matlab code reads in the input data from a SQLite database. This input data is stored in two tables in the database according to the structure presented (see Section 4.1). Afterwards, the Matlab program executes the described steps of our approach up to the identification of control logics (see Figure 2). The resulting formal simulation model is saved as an Extensible Markup Language (XML) file. The file structure is based on the Core Manufacturing Simulation Data (CMSD) model (Bergmann and Straßburger 2015). Tecnomatix Plant Simulation reads this file and automatically builds the simulation model according to it.

The validation of the functions for automated model development and parameterization were tested on a laboratory scale. For this purpose, a fictional production system was set up in Tecnomatix Plant Simulation. This model generates the described input data during its execution. Criterion for the validation of the presented approach is the extent to which the approach can automatically and correctly reproduce the material flow model, the parameterization of material flow elements and the control policies of the given simulation model on the basis of the input data.

For the validation we used the simulation model shown in Figure 4. We set up nine scenarios, each of which differed in the routing strategy applied to processing station A. The tested routing strategies are: (1) defined successor, (2) linear sequence, (3) cyclic, (4) cyclic sequence, (5) percent, (6) / (7) max. / min. contents, (8) / (9) max. / min contents blocking. For each scenario we executed the simulation with a fixed shift calendar with three shifts over a simulation period of 14 days.

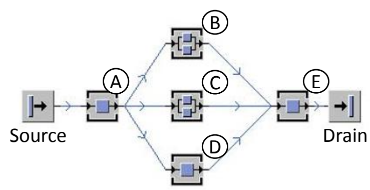

Figure 4: Structure of test scenarios in Tecnomatix Plant Simulation. Processing stations A, D and E have fixed processing times (A: 1:00min, D: 1:00min, E: 0:30min). The processing times of B and C are uniformly distributed (B: 1:00min - 2:00min, C: 2:00min - 4:00min). In each scenario a specified routing rule is applied at the part exit of processing station A.

Using the generated data as a input, our approach correctly reconstructs the material flow elements and their connections. The shift calendar is also correctly recognized. The approach provides valid values for process and setup times. The distributions of processing time at stations B and C are reconstructed correctly. All control policies are recognized correctly. This first results show that our approach works well on a laboratory scale. However, further tests with variations of the simulation period show, that the approach fails to identify the shift calender correctly for simulation periods below 14 days. Further, a correct reconstruction of routing rules and the processing time distributions requires at least a 2 day simulation period.

6 DISCUSSION

With this approach, we specifically address the challenges in simulation projects that lead to long development times. In the first step, we give the user a precise definition of the input data required for our approach. The input data is based on the minimal requirements for manufacturing data acquisition systems derived from the literature. In this way, we ensure that the required data is available in various proprietary IT systems for manufacturing data acquisition and hardly requires any further manual preparation by the simulation expert. We facilitate the steps of identifying the required data sources and reduce the effort involved in data preparation. This distinguishes our approach from the state of the art of automated IDM, which specifically focus on a simplified connection of data sources in general and on the support of data processing. This allows these approaches greater flexibility using several data sources and in identification of parameters, but this comes at the expense of the degree of automation.

The automation of the model development phase eliminates the need to record the system complexity by the simulation expert, as the material flow elements, material flow relationships and control policies are automatically derived by our approach. Here, our approach extends the state of the art for automated model generation by combining automated identification of material flow elements and identification of control logic.

We have implemented the approach prototypically, tested and validated it on a laboratory scale. First tests show promising results. The approach enables the automated development and parameterization of simulation models according to the described specifications.

Even if the first implementation of the approach looks promising, a critical discussion is needed. In the area of input data, it should be noted that although attempts have been made to keep the data requirements low, they may not always be met in practice. For example, some of the processing stations in real production systems might not be equipped with the necessary tracking points to record data. If this is the case, the material flow can currently only be reconstructed incompletely. However, information about these sections can still be extracted from the data. Methods to use this information need to be developed.

Furthermore, the approach has so far only been tested and validated on a laboratory scale. In real systems, therefore, a different quality of input data can be expected. For example, part routes can be falsified by manually infiltrating and exfiltrating parts, or deviations from shift calendars can occur by working through pauses. In this environment, the developed methods still have to be tested for their stability and extended if necessary.

When reconstructing the material flow model, the approach so far only distinguishes between process stations and buffers. A further detailing around further logistic and production-related material flow elements is necessary for a valid modeling of real systems.

7 CONCLUSION AND OUTLOOK

In this paper we describe a new approach to automated model development and parameterization of material flow simulations using tracking and tracing data. A first prototypical implementation and a validation on a laboratory scale show promising results. Our approach's main contribution to the research topic of automated model generation and IDM is: (1) To be a holistic approach, automating not only parametrization but also model generation. (2) To reduce efforts in finding and integrating several data sources by only using common data sources, setting minimal requirements to the input data, and explicitly describe the structure of the data needed.

By automating time-consuming steps, the effort and development time in simulation projects can be shortened. Further, the automation leads to more accurate parametrization, as the input data is more up-to-date, which increases the quality of decision-making supported by simulation. Possible use cases are reconfigurations of the production resources or the evaluation of alternative production programs to cope with fluctuating demands.

As next steps and further research topics we identify the three following points: (1) The extension of the methods for the automated identification of more detailed material flow elements. We try to achieve this, by modeling the characteristic differences between elements. For example a conveyor band can be differentiated from a buffer by a fixed retention time of through going parts following a FIFO logic. (2) The development and extension of methods for handling data of lower quality from real production system. This could be realized by integrating the automated methods into an assistance system in order to complement missing information with expert knowledge by means of manual input. (3) Test and validation of the approach with data from a real production system.

REFERENCES

Aufenanger, M., A. Blecken, and C. Laroque. 2010. "Design and Implementation of an MDA Interface for Flexible Data Capturing". *Journal of Simulation* 4:232–241.

Barlas, P., and C. Heavey. 2016. "KE Tool: An Open Source Software for Automated Input Data in Discrete Event Simulation Projects". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 472–483. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Bergmann, S. 2013. *Automatische Generierung adaptiver Modelle zur Simulation von Produktionssystemen*. Ph.D. thesis, Ilmenau, Techn. Univ., Ilmenau. http://www.db-thueringen.de/servlets/DocumentServlet?id=23106, accessed 23rd July 2019.

Bergmann, S., and S. Straßburger. 2015. *On the Use of the Core Manufacturing Simulation Data (CMSD) standard: Experiences and Recommendations*. Ilmenau: Univ.-Bibliothek TU Ilmenau.

Charpentier, P., and A. Véjar. 2014. "From Spatio-Temporal Data to Manufacturing System Model". *Journal of Control, Automation and Electrical Systems* 25:557–565.

Dangelmaier, W. 2001. *Fertigungsplanung*. Berlin, Heidelberg: Springer Berlin Heidelberg.

ElMaraghy, H. A. 2008. *Changeable and Reconfigurable Manufacturing Systems*. Luxemburg: Springer Science & Business Media.

Fishman, G. S. 2001. *Discrete-Event Simulation: Modeling, Programming, and Analysis*. Springer Series in Operations Research. New York, NY: Springer.

Hippenmeyer, H., and T. Moosmann. 2016. *Automatische Identifikation für Industrie 4.0*. Berlin, Heidelberg: Springer.

Lödding, H. 2016. *Verfahren der Fertigungssteuerung*. 2., erw. ed. Berlin, Heidelberg: Springer Berlin Heidelberg.

Mayer, G., and M. Mieschner. 2017. "Industry 4.0 - Chance or Risk for Material Flow Simulations". *Simulation in Produktion und Logistik* 2017.

Perera, T., and K. Liyanage. 2000. "Methodology for Rapid Identification and Collection of Input Data in the Simulation of Manufacturing Systems". *Simulation Practice and Theory* 7:645–656.

Selke, C. 2005. *Entwicklung von Methoden zur automatischen Simulationsmodellgenerierung*. Ph.D. thesis, Techn. Univ. München, München.

Skoogh, A., B. Johansson, and J. Stahre. 2012. "Automated Input Data Management: Evaluation of a Concept for Reduced Time Consumption in Discrete Event Simulation". *Simulation* 88:1279–1293.

Smith, C. R. 2015. *The Programmatic Generation of Discrete-Event Simulation Models from Production Tracking Data*. Master Thesis, Brigham Young University, Provo, Utah. https://scholarsarchive.byu.edu/cgi/viewcontent.cgi?referer=https://www.google.de/&httpsredir=1&article=6828&context=etd, accessed 23rd July 2019.

Vasyutynskyy, V., A. Gellrich, K. Kabitzsch, and D. Wustmann. 2010. "Analysis of Internal Logistic Systems based on Event Logs". In *2010 IEEE 15th Conference on Emerging Technologies & Factory Automation (ETFA 2010)*, 1–8. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

VDI 2016a. "Manufacturing Execution Systems (MES)". Last modified October, 2016. https://www.beuth.de/de/technische-regel/vdi-5600-blatt-1/251310997, accessed 23rd July 2019.

VDI 2016b. "Modellierung und Simulation - Modellbildungsprozess". Last modified May, 2016. https://www.vdi.de/richtlinien/details/vdi-4465-blatt-1-modellierung-und-simulation-modellbildungsprozess, accessed 23rd July 2019.

Yang, J., W. Wang, and P. S. Yu. 2003. "Mining Asynchronous Periodic Patterns in Time Series Data". *IEEE Transactions on Knowledge and Data Engineering* 15:613–628.

AUTHOR BIOGRAPHIES

MICHAEL MILDE received his M. Sc in Physics from Technical University of Munich (TUM) in 2016. He joined the Institute for Machine Tools and Industrial Management in 2017 where he works as a researcher in the department for production management and logistics. His research interest are in data mining, discrete event simulation and global manufacturing networks. His e-mail address is michael.milde@iwb.mw.tum.de.

GUNTHER REINHART is Professor of Industrial Management and Assembly Technology at TUM. He studied mechanical engineering with a focus on design and development until 1982 and completed his doctorate in 1987 at the Institute for Machine Tools and Industrial Management (iwb) of the Technical University of Munich. From March 2002 to February 2007, Prof. Reinhart was on leave from his activities at the iwb and took over the position of Chief Technology and Market Officer at IWKA Aktiengesellschaft in Karlsruhe. Since January 1, 2009, Prof. Reinhart has also been head of the Fraunhofer IWU Project Group for Resource-Efficient Mechatronic Processing Machines (RMV) in Augsburg, which since July 1, 2016 has been integrated into the newly founded Fraunhofer Institute for Foundry, Composite and Processing Technology (IGCV). His e-mail address is gunther.reinhart@iwb.mw.tum.de.