Virtual Stochastic Sensors for Reconstructing Job Shop Production Workflows

Claudia Krull, Graham Horton
Department of Computer Science
Otto-von-Guericke University Magdeburg
Magdeburg, Germany
e-mail: claudia.krull@ovgu.de,
graham.horton@ovgu.de

Berend Denkena, Barbara Dengler
Institute of Production Engineering and Machine Tools
Leibniz University Hannover
Hannover, Germany
e-mail: dengler@ifw.uni-hannover.de

***Abstract*— Virtual stochastic sensors (VSS) can reconstruct the system behavior of partially observable systems that may contain concurrent non-Markovian activities. These systems can for example occur in industrial production. In this paper, we apply VSS to reconstructing workflows in a job shop. The example application was designed in cooperation with logistics experts to resemble a real job shop. In order to deal with probabilistic decisions in the workflows, we extended Hidden non-Markovian Models to include immediate transitions and modified the corresponding Proxel-based behavior reconstruction algorithm accordingly. Experimental data was acquired using a setup including a printed layout, RFID sensors and a central data collection unit. We were able to reconstruct the actual workflows from the acquired sensor data, which for the first time shows an application of VSS to real measurement data. This first practical application, and the extension of the modeling paradigm takes us forward to our goal of a realistically applicable method for behavior reconstruction based on partial system observations.**

Keywords-discrete stochastic systems; hidden non-markovian models; virtual stochastic sensors; behavior reconstruction; workflow modelling and simulation; analytical simulation

I. INTRODUCTION

Virtual stochastic sensors (VSS) have been introduced as a method to reconstruct the system behavior of partially observable processes, for example in industrial production. In the examples used so far we assumed that these processes involved the processing of goods on assembly lines. Job shops present another kind of production processes, which bear several challenges.

Firstly, in a job shop production it is difficult to completely monitor the process via sensors due to the manual handling and transportation of goods between workplaces and storage areas. Secondly, these manual handling processes also complicate the placing of sensors for monitoring, since it is not clear if and where exactly the production items will pass a certain neuralgic point. Thirdly, the processing workflow is usually not as well-defined as in an assembly line scenario. Usually there exist one or more standard workflows, which are executed with certain variations or optional process steps, depending on the specific job requirements. In order to analyze or optimize the workshop processes, exact measures are needed, e.g. for utilization or machine availability. Therefore, there is a great interest in reconstructing the complete workflow for specific jobs based on sensor data, in order to determine relevant performance figures.

From a modeling perspective, the workflows in a job shop have one major difference compared to assembly line scenarios. In the latter, the same or similar activities are always executed in the same order. In a job shop production process, the possible variations to the standard workflows can be more diverse, and are caused by specific requirements of single jobs. These possible variants can only be feasibly modeled by probabilistic decisions. Our VSS analysis method is currently not able to handle such model elements as probabilistic decisions. We therefore need to extend our modeling paradigm of Hidden non-Markovian Models (HnMM) by immediate transitions with probabilities and modify the behavior reconstruction algorithms accordingly.

In this paper an example of a job shop production process is used, which was designed in cooperation with logistics experts to resemble a real job shop. Experimental data was acquired using a setup including RFID sensors and a central data collection unit. HnMM were modified to include immediate transitions in order to handle probabilistic decisions, which increases the class of models that VSS are applicable to. We were able to reconstruct the actual workflows from the acquired sensor data, which for the first time shows an application of VSS to real measurement data. This again takes us forward to our goal of a realistically applicable method for behavior reconstruction based on partial system observations.

II. RELATED WORK

A. *Virtual Stochastic Sensors*

Virtual stochastic sensors (VSS) [5] are a method to reconstruct the behavior of partially observable discrete stochastic systems. Based on system output protocols with time stamps they can determine the probability of a particular output protocol or determine likely system behavior to have produced the observed output. They are related to so-called virtual sensors (VS) [3], where a quantity of interest is not directly measurable, but can only be calculated based on its relationship to other measurable system quantities. In VSS, the relationship between the measurable and the interesting quantities cannot be described analytically, but has a stochastic nature.

Hidden non-Markovian Models (HnMM) are used to represent these partially observable systems [4]. HnMM can

978-0-7695-5073-2/13 $26.00 © 2013 IEEE
DOI 10.1109/EUROSIM.2013.57

represent processes containing multiple concurrent non-Markovian activities. It is assumed that the completion of some of these activities causes the emission of an observable output signal, which can be ambiguous; i.e. multiple activities may emit the same signal, and the same activity may emit different signals with different probabilities.

Since HnMM were originally based on Hidden Markov Models, the central questions addressed by VSS are two of the main tasks common to all hidden models. The Evaluation task is to determine the overall probability of a given output protocol for a given model specification. The Decoding task is to reconstruct the most likely system behavior of a given model to have produced a given output protocol.

The state space-based Proxel simulation method can be used to reconstruct the possible system behavior (path) to have produced a given protocol of output signals [5]. Using Proxels we are also able to determine the probability of a system path and the overall probability of a protocol. Several modifications of the method exist that address different HnMM modifications.

B. *Proxel-based Simulation*

A Proxel is a probability element, which is the core computational unit of the state space-based analysis method [2][7]. The Proxel simulation algorithm traverses and quantifies all possible system development paths in discrete time steps. One Proxel contains all the information needed to determine possible future behavior of the system: the discrete system state, the age of all relevant non-Markovian transitions and the probability of that combination. This enables the exact treatment of continuous non-Markovian distribution functions without having to deal with systems of partial differential equations.

The general procedure of the iterative algorithm is the following: Starting with one or more initial Proxels at simulation time 0, the Proxels of one time step are processed one by one. For each Proxel's discrete state the possible follow-up states are determined and the corresponding transition probability is computed based on the transition ages. Then the child Proxels are created and stored for the next time step. This procedure is repeated for all time steps until a predefined simulation time or some other termination criterion is reached. The algorithm specifically assumes that the model spends at least one full time step in each state before leaving that state again, an error made consciously to gain such an easy algorithm.

This procedure, however, does not take into account immediate transitions, which change a model's state instantaneously upon becoming enabled. An extension to include immediate transitions in the Proxel algorithm was introduced in [6]. However, the paper proposed to handle immediate transitions directly upon becoming enabled, not even producing the vanishing states that they result in. The proposed solution is not flexible enough for the current application, since the workflow models might contain several successive decisions for optional process steps, which would result in several consecutive immediate transitions in the model. We therefore need to find a more general procedure to handle immediate transitions, which also fits into the regular step of processing a Proxel.

C. *Job Shop Production Processes*

A job shop is a small manufacturing system designed to handle small or single order products, which differ from each other completely or by customizable features. The order and size of the jobs is very flexible and determined by customer demand. In job shops, the machines are usually clustered according to their technological process and different jobs may require different sets of these processes to be completed. The goods are moved manually between the different shops and intermediate storage areas as necessary. Due to the high flexibility compared to continuous flow manufacturers, scheduling and analysis of job shops is very difficult, according to literature. Most job shops are small and medium sized businesses, and evolved and adapted to customer demands as necessary. Due to increasing internationalization and market competition, a structured evaluation and optimization of the manufacturing processes becomes more interesting in order to stay competitive. Therefore, we chose job shops as our application scenario, since the need for a flexible process analysis method is apparent.

III. The Application Scenario

A. *Example Job Shop Production Scenario*

An example job shop scenario was developed based on an actually existing workshop. The layout in Figure 1 contains several different workplaces with diverse machines (depicted using standard machine illustrations) and storage areas. The marked areas identify the distinct workplaces (1, 2, 3) and storage areas (A, B, C) as well as the entrance and exit point (E), which were involved in the workflows to be monitored in the later experiment.

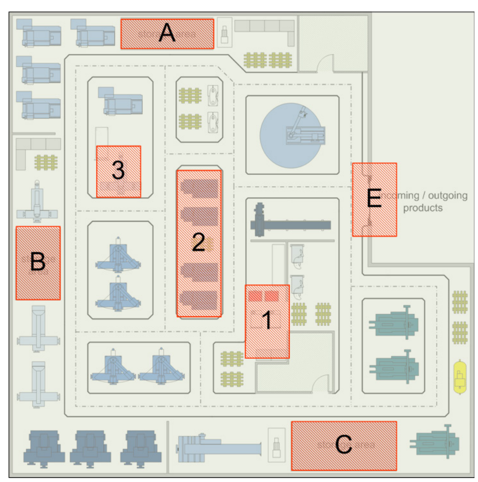

Figure 1. Example Workshop Layout with Areas of Interest Marked

The workflows that we defined have their equivalent in the existing prototypical workshop. There are three different workflows with two variations each, involving the locations marked in Figure 1. The workflows are the following:

E-C-1-E
E-C-1-B-E
E-A-2-3-B-E
E-A-2-B-3-E
E-A-1-2-B-E
E-A-1-B-2-E.

Each of the workflows includes one or two processing steps, and at least one storage step. Since the experiment setup planned for data collection required manual handling and stopwatch-guided virtual processing times, we scaled the real system times to seconds as our model time unit. However, the method is in no way dependent on such short time spans. When assuming the time unit to be minutes, hours or days, the scenario becomes realistic for a real job shop production, and the results are directly applicable.

The durations of the three processing steps performed at the workplaces marked in Figure 1 were assumed to be governed by the following Uniform distributions:

Step 1 ~ Uniform (20s, 30s)
Step 2 ~ Uniform (10s, 20s)
Step 3 ~ Uniform (15s, 25s)

This assumption is reasonable also for other job shop environments, since the data obtained from process planning systems will usually be sparse and therefore best represented by this simple distribution type.

The duration of a transport within the workshop as well as the times to enter or leave the workshop were assumed to be normally distributed with a mean value of three seconds and a standard deviation of two seconds. The time spent in storage areas was not restricted.

We had four RFID reading sensors available to distribute among the neuralgic points of our workshop. The locations chosen for these sensors were the entrance/exit E point, the storage areas A and B, as well as the workplace for Step 1. One sensor can roughly cover the complete physical area of the process step at its location, so that no items will be missed. Furthermore, all products were marked with distinct RFID tags and the assignment of the tag to a certain job is known. One RFID sensor collects the information when a certain tag enters its range, and when the range is left.

B. *Modelling the Process from the Workflows*

In order to analyze this scenario using VSS behavior reconstruction, we needed to turn these workflows into one or more suitable HnMM specifications. We decided to analyze the workflows of the items individually, since we knew that an RFID tag was associated to a specific job for a known time period, and the beginning and end of a workflow are clearly marked by passing the entrance/exit sensor. Apart from being far more complex, a model reflecting the processing of multiple items in parallel would not have yielded any benefit, given our real system. If a target system contained limited resources such as transport capacity or local storage areas besides workplaces, then the inclusion of multiple items in the model might become necessary.

We decided to compare two different approaches to solving the problem at hand. As the first option we combined all six workflows in one HnMM, reflecting the complete workshop process to be monitored. Decoding is then used to find the most likely path through the model, which hopefully would correspond to one of our workflows.

The second option was to specify one HnMM for each of the workflows, each only representing one possible path through the system. Using Evaluation we then determine the probabilities of the different models for the same protocol and select the most likely workflow model. Due to the lack of space, we will not describe the models of this simpler option here, but only refer to it in the experiments section.

We used augmented stochastic Petri nets (ASPN) [1] as higher level description of the HnMM to represent the job shop workflows. An excerpt of the graphical representation of the ASPN can be seen in Figure 2. Rectangles represent process activities or state transitions, circles the process steps, solid arrows the workflow direction and dashed arrows the signal outputs. The process contains several decisions, which are represented as immediate transitions (solid bars). Without loss of generality we assumed multiple competing immediate transitions to have equal probability. The transitions that make a job enter or leave the area of a certain sensor are augmented with symbols for the firing of that sensor (dashed arrows). Entering and leaving a sensor area results in the same symbol, leading to four output symbols (1 at E, 2 at A, 3 at B, 4 at Workplace 1). Each signal is emitted by the corresponding transition with the probability of 1.0.

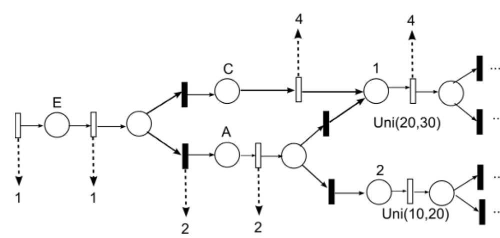

Figure 2. ASPN Excerpt of all Workflows Combined in one Model

The state space of the system, as depicted in Figure 3 contains the possible distinct locations (circles) of a single job in the workshop. The transitions (arrows) between them signify the different possible workflows. Their annotations with the process step durations or decision probabilities were left out for reasons of clarity. The state space contains several vanishing markings (dashed circles), due to the immediate transitions in the ASPN. In a GSPNs state space, one can remove vanishing markings by multiplying the outgoing probabilities with the transition rates leading to the markings [8]. However, our model contains non-Markovian transitions, which lead to these vanishing markings. Therefore, we cannot eliminate the vanishing markings and retain them in the discrete model state space.

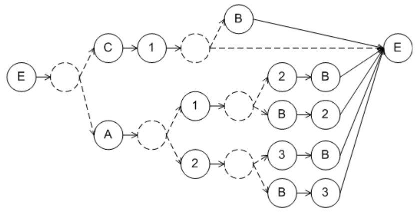

Figure 3. Discrete State Space of ASPN Containing all Workflows

The resulting discrete model state space is different from the ones we have analyzed in previous research, which contained only tangible states, with non-zero state holding times. Therefore we have to modify our analysis method slightly, in order to be able to analyze models containing both vanishing and tangible states.

C. *Modifying the Analysis Method*

The Proxel method is able to incorporate immediate transitions. However, as mentioned in Section II.B, the method proposed in [6] is not flexible enough for our purposes. We therefore propose the following procedure for handling immediate transitions in the Proxel processing step:

- If there are no immediate transitions enabled in the Proxels discrete state, process it as usual and create child Proxels for the next time step.
- If there are immediate transitions enabled, disregard all other transitions and determine the probabilities of the immediate state changes. Create child Proxels for the **current** time step.

This will result in these child Proxels to be processed in the same time step as their parents, and thus they will behave as if no time had elapsed since the entering of the state. This procedure also allows for multiple successive immediate transitions in the model to be all processed in the same time step. The procedure introduces a small error, since effectively the immediate transitions take effect one item step too late, but the error is not cumulative for successive immediate transitions and therefore acceptable. The overhead of the approach is almost non-existent and the modifications necessary to the algorithm are very small.

One further issue is the handling of time steps where output signals were emitted. Here we needed to modify the data structure such that we could keep track of whether the emission was already covered by one of the possibly multiple successive transitions in one time step. Thus we can make sure that one and only one transition firing within one time step emitted the signal. With the analysis method adapted, we were able to do behavior reconstruction for the models defined in Section III.B.

IV. EXPERIMENTS

A. *Data Acquisition*

Because of privacy issues and technical difficulties it was not possible to collect experimental data in the real job shop environment. Therefore we used an experiment setup that was based on the layout from Figure 1 printed in the dimensions of (1mx1m), fixed to a plywood board. The sensors were placed beneath the board in order to not disturb the workflow. The sensor data was collected in a central unit containing a database. This essentially synchronized the sensor readings and collected the sensor data in a suitable database format. The time stamps of entering and leaving a sensor zone were recorded to an accuracy of one second, due to technical restrictions. The RFID tags were attached to Lego cubes and handled as single order.

The manual data generation process was the following:

- A certain tag is selected and the desired workflow is chosen.
- The tag is then placed on the entrance/exit location, for a small amount of time.
- The tag is advanced according to the workflow until the workflow is completed, and the tag is again placed on the entrance/exit location.
- The tag is placed in a workplace area for the time period prescribed by the processing time, which was checked by a stopwatch.
- The tag is placed in a storage place for an unspecified amount of time.

A picture of the data acquisition process can be seen in Figure 4. The workflows were executed both individually and in parallel to simulate a real job shop environment. In order to check the accuracy of the behavior reconstruction, an experiment protocol was written, containing the order and time of the execution of the different workflows.

Figure 4. Experiment Setup with Data Collection Unit and Tags

An excerpt of the raw output of the data acquisition phase can be seen in Figure 5. The table includes the following fields: Id (unique DB key), TimeStamp (time of DB entry), Duration (period within sensor range), OptiBoxID (sensor number), TagID (unique RFID tag identifier), StartDate (when sensor range was entered), EndDate (when sensor range was left).

Id	TimeStamp	Duration	OptiBoxId	TagID	StartDate	EndDate
4830	22.04.2013 14:47	2	3	62109	2013-04-22 14:47:43.000	2013-04-22 14:47:45.000
4828	22.04.2013 14:47	0	2	62109	2013-04-22 14:47:19.000	2013-04-22 14:47:19.000
4829	22.04.2013 14:47	2	3	62136	2013-04-22 14:47:11.000	2013-04-22 14:47:13.000
4827	22.04.2013 14:47	2	3	62111	2013-04-22 14:47:07.000	2013-04-22 14:47:09.000
4823	22.04.2013 14:46	8	1	62111	2013-04-22 14:46:57.000	2013-04-22 14:47:05.000
4824	22.04.2013 14:46	20	2	62136	2013-04-22 14:46:49.000	2013-04-22 14:47:09.000
4826	22.04.2013 14:46	2	3	62122	2013-04-22 14:46:45.000	2013-04-22 14:46:47.000
4821	22.04.2013 14:46	12	1	62136	2013-04-22 14:46:31.000	2013-04-22 14:46:43.000
4820	22.04.2013 14:46	18	1	62122	2013-04-22 14:46:25.000	2013-04-22 14:46:43.000
4822	22.04.2013 14:46	38	4	62111	2013-04-22 14:46:17.000	2013-04-22 14:46:55.000
4825	22.04.2013 14:46	72	1	62109	2013-04-22 14:46:05.000	2013-04-22 14:47:17.000
4818	22.04.2013 14:46	22	4	62122	2013-04-22 14:45:49.000	2013-04-22 14:46:11.000
4815	22.04.2013 14:45	12	2	62122	2013-04-22 14:45:34.000	2013-04-22 14:45:46.000

Figure 5. Excerpt of Raw Data Obtained from Sensors

B. Data Preprocessing

The raw data had to be preprocessed to enable reconstruction of the paths of single jobs through the system. We therefore transformed the real time stamps into a suitable integer format by converting the clock time to seconds. Since we only had a limited supply of RFID tags, some were used several times. We therefore had to separate the distinct workflows conducted with the same tag, which was done by taking the sensor readings at the entrance/exit points as beginning and end of a trace. In each trace, we then reduced the absolute values of the time stamps to below 1000, where possible. This does not change anything in the results, except for the absolute probability of a trace, which was not relevant in the application scenario. However, this reduction of time stamps reduced the computational effort considerably.

The experimental data collection consisted of two hours on two different occasions, resulting in 69 distinct workflows. The later analysis showed that 52 of these were using correct times and sequences of the process steps. The others were not usable due to manual errors in the data collection process. Some contained sensor fragments, when an item was moved between two locations, and in the transport process entered another sensors area, but these workflow traces were kept intentionally, and the fragmental readings were removed, since such problems will likely also occur in a real world sensor setting.

This preprocessing of the raw data was done manually for the experiments conducted here. However, the process can easily be automated.

C. Path Reconstruction by Decoding Experiment

In a first experiment we test path reconstruction using the model containing all workflows and the Decoding method. This means we use a single trace as our model input. The method output contains the possible system paths and their respective probabilities. We then choose the most likely path through the system as our winner and compare it to the actual workflow that was used to generate the trace. If there are multiple paths possible, we will alternatively compare the cumulative probabilities of the distinct system paths and compare them to the original workflow.

Our example trace is the following:

timestamp	symbol
815	1
817	1
819	2
829	2
877	3
891	3
893	1
909	1

Figure 6. Preprocessed Protocol of Sensor Readings for Example Workflow

When considering the order of the output symbols (1 for E, 2 for A, 3 for B), this trace can be identified as one of two possible workflows (E-A-2-3-B-E or E-A-2-B-3-E). The behavior reconstruction of the path yielded 2776 possible paths. An excerpt of these paths can be seen in Figure 7. The first two columns correspond to the logarithmic and the normal probability of the path, the following columns are pairs of state change timestamps and system states. The first path in the table is also the one with the highest probability. All reconstructed possible system paths correspond to the workflow E-A-2-3-B-E. This was also the one used to generate the example protocol.

log_prob	prob	t	s	t	s	t	s	t	s	t	s	t	s	t	s	t	s	t	s	t	s	t	s	t	s
-117.88573	6.35E-52	815	Enter	817	TA	819	PL_A	829	T2	832	AP_2	851	T3	854	AP_3	874	TB	877	PL_B	891	TE	893	Exit	909	End
-117.93225	6.06E-52	815	Enter	817	TA	819	PL_A	829	T2	832	AP_2	850	T3	853	AP_3	874	TB	877	PL_B	891	TE	893	Exit	909	End
-117.951246	5.95E-52	815	Enter	817	TA	819	PL_A	829	T2	833	AP_2	849	T3	853	AP_3	874	TB	877	PL_B	891	TE	893	Exit	909	End
-118.159839	4.83E-52	815	Enter	817	TA	819	PL_A	829	T2	832	AP_2	847	T3	849	AP_3	873	TB	877	PL_B	891	TE	893	Exit	909	End
-118.209901	4.59E-52	815	Enter	817	TA	819	PL_A	829	T2	833	AP_2	851	T3	855	AP_3	875	TB	877	PL_B	891	TE	893	Exit	909	End
-122.459108	6.56E-54	815	Enter	817	TA	819	PL_A	829	T2	836	AP_2	850	T3	852	AP_3	871	TB	877	PL_B	891	TE	893	Exit	909	End
-127.798887	3.15E-56	815	Enter	817	TA	819	PL_A	829	T2	836	AP_2	846	T3	853	AP_3	870	TB	877	PL_B	891	TE	893	Exit	909	End

Figure 7. Possible Generating Paths of Example Workflow Protocol

We repeated the experiment evaluation for all 52 protocols not containing data collection errors. Every protocol could be reconstructed to its original workflow.

Depending on the protocol, the number of possible paths varied from less than twenty to several thousands. The probabilities of the 2776 different paths of our example protocol can be seen in Figure 8.

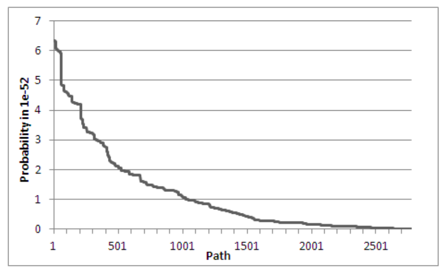

Figure 8. Plot of Path Probabilities in Descending Order for Example Protocol

Only one type of workflow (E-A-2-B-3-E) yielded reconstruction results with different possible workflows. However one of the reconstructed workflows did not correspond to the originally defined ones, but was a possible path in the resulting model. However, even here the actual generating workflow was the one with more probability.

The experiment shows that we were able to reconstruct the different workflows from the sensor data using the model containing all possible workflows. The runtime was in all cases under one minute and therefore not further examined.

This good result of the behavior reconstruction is possibly due to the clear distinction of the workflows by the four sensors. The three distinct types of workflows can be distinguished only by the order of the sensor outputs.

D. Path Reconstruction by Evaluation Experiment

In our second experiment, we used six separate models for the different workflows. We used the 52 protocols from the first experiment and calculated the evaluation probability for every protocol/model combination. The results were very similar to the first experiment; we were also able to reconstruct the true generating workflow for every sensor protocol. Furthermore no alternative paths were reconstructed, not corresponding to valid workflows, since these were just not possible in our six models.

E. Benchmark When Removing Data

Since the workflows could easily be distinguished in the experiments described in the previous sections, we modified the setting by removing the sensor at workplace 1 from the model and protocols. As a result, the four workflows containing both storage areas A and B would produce the same order of sensor readings. These are the ones considered further in this experiment. We then conducted the decoding and evaluation experiments with the modified models and protocols. The results of both experiments were the same; both options reconstructed the same path for each protocol. The results can be seen in the confusion matrix shown in Figure 9. The table shows how many of the different protocols produced using one workflow were reconstructed to every possible workflow.

	A1B2X	A12BX	A2B3X	A23BX
A1B2X	9			
A12BX		2		5
A2B3X			9	
A23BX				10

Figure 9. Confusion Matrix when Omitting Sensor at Workplace 1

The table shows that by removing one sensor from the model we produced ambiguity among the different workflows. Even though we were able to reconstruct 30 of the 35 workflows considered. The experiments show that in job shop environments where details of the possible workflows are known and the different production steps can be distinguished by their duration, virtual stochastic sensors are able to reconstruct the workflows based on protocols produced by sensors at neuralgic points in the workshops.

V. CONCUSIONS AND OUTLOOK

A. Summary of Results

The paper demonstrates the application of virtual stochastic sensors for behavior reconstruction in a job shop. Experimental data was gathered using a realistic workshop layout with real sensor equipment and manual execution of workflows. The successful behavior reconstruction from the resulting data shows that virtual stochastic sensors can help in analyzing workflows in a job shop environment.

Furthermore did the application example require the inclusion of probabilistic decisions in the model, resulting in vanishing markings in the discrete models state space. We were able to modify our behavior reconstruction method to suit this model type. Thereby, we were able to extend the model class for which we can perform behavior reconstruction from partial observations to systems which include immediate transitions in their process representation.

B. Result Discussion

The application example was modeled to closely resemble a real workshop. The experimental setup was also realistic, even though the time and space scale was reduced to enable manual data generation. Therefore we can assume that the results of the paper are more generally applicable. We have thus shown a first successful application of virtual stochastic sensors to a real problem.

The extension of HnMM by immediate transitions also increases the applicability of VSS to real models. As stated, in real job shop environments we often deal with alternative or optional process steps and multiple possible orders. Many other real systems can be modeled more intuitively when allowing for immediate transitions representing choices or conditions. Therefore, this extension, even though it involved little modification in the algorithm, largely increased the practical applicability of virtual stochastic sensors.

C. Future Research

Further experiments have to be conducted using larger and more ambiguous workflow sets. Experiments should also test the methods' behavior when including faulty sensors, producing missing or erroneous sensor readings. This will require an adaption of the workflow models to include different sensor output probabilities. Earlier research has already demonstrated the ability of virtual stochastic sensors to correctly reconstruct more ambiguous data sets. Therefore, we expect the method to be able to cope with such challenges, even though losing result accuracy.

ACKNOWLEDGMENT

The results presented have been developed within the scope of the project "OptiBox - Optimization of production systems through self-learning and mobile analyses" (Grant Number: 02PK3029) funded by the Federal Ministry for Education and Research (BMBF).

REFERENCES

[1] R. Buchholz, „Conversive Hidden non-Markovian Models“ PhD thesis at the faculty of computer science at the Otto-von-Guericke-University Magdeburg, Mai 2012.

[2] G. Horton, “A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times.” Proc. of European Simulation Symposium, 2002. SCS European Publishing House.

[3] P. H. Ibargüengoytia and A. Reyes, “Constructing Virtual Sensors Using Probabilistic Reasoning” In MICAI 2006: Advances in Artificial Intelligence, LNCS Vol. 4293, 2006, pp 218-226.

[4] C. Krull and G. Horton, “Hidden Non-Markovian Models: Formalization and Solution Approaches” Proc. of 6th Vienna International Conference on Mathematical Modelling, 11th-13th February 2009, Vienna, Austria.

[5] C. Krull, R. Buchholz and G. Horton:, “Virtual Stochastic Sensors: How to gain Insight into Partially Observable Discrete Stochastic Systems” Proc. of the 30th IASTED MIC, 2011, Innsbruck. Austria.

[6] S. Lazarova-Molnar and G. Horton, “Proxel-Based Simulation of Stochastic Petri Nets Containing Immediate Transitions, ” On-Site Proceedings of the Satellite Workshop of ICALP 2003 in Eindhoven, Netherlands. Forschungsbericht Universität Dortmund. Dortmund 2003.

[7] S. Lazarova-Molnar, „The Proxel-based Method: Formalisation, Analysis and Applications“ PhD thesis at the faculty of computer science at the Otto-von-Guericke-University Magdeburg, November 2005.

[8] M. A. Marsan, G. Conte, and G. Balbo. 1984. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. ACM Trans. Comput. Syst. 2, 2 (May 1984), 93-122.