SIMULATION OF HORIZONTAL AND VERTICAL INTEGRATION IN DIGITAL TWINS

Stefan Haag and Carlo Simon
Fachbereich Informatik
Hochschule Worms
Erenburgerstr. 19, 67549 Worms, Germany
E-Mail: {haag,simon}@hs-worms.de

KEYWORDS

Horizontal and Vertical Integration, IoT, Cyber-physical System, Digital Twin, Simulation, Petri nets

ABSTRACT

Horizontal integration describes the linkage of processes to simplify the flow of materials and information between different corporations. Vertical integration describes the connection or merging of processes between top and shop floors inside the very same organization to enable comprehensive possibilities for optimization, allowing – virtually as side effect – for the customized production of batch size one orders.

Both fields offer interesting issues, for example how to gain insight into what kind of effects a possible transformation could have prior to conducting said transformation – maybe assembly of a new production line, reengineering of a supply chain or managing the change of a business process due to new market constraints. Alterations naturally come at a price, thus, testing the consequences by implementation is nearly always infeasible.

To lessen the cost of change, transformations can be analyzed using digital twins – by modelling real world machinery, work pieces and processes, then simulating them and their interaction.

The authors present a possibility to implement digital twins in arbitrary magnitude and level of detail. A web-based specification and simulation environment allows for modelling both high-level business processes and field level as well as activities across companies' borders, comprising everything between.

INTRODUCTION

The total or partial replacement of analogue processes by digital, computer-manageable processes are a main cause of the industrial revolution that can be observed today (Wolf and Strohschen 2018). It leads to disruptive changes in several industries where more and more decisions are made by algorithms instead of employees. As a precondition, real world information must be available in an integrated model in order to simulate possible future paths before conducting the best one.

Two integration scenarios are discussed as shown in figure 1: while horizontal integration means the integration of business processes along supply chains, vertical integration means to connect the information systems from the top floor (i.e. the business processes automated by an ERP system) to those of the shop floor (i.e. the IT systems for controlling the machines in production).
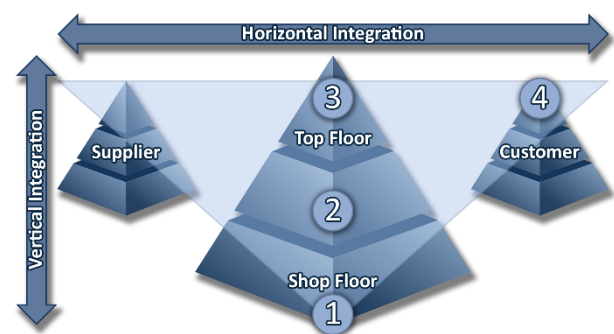

Figure 1: Integration of Processes across Companies

In this paper, possible implementations on the numbered items are introduced. For item 1 a PLC-like approach to simulate the field level is used, while an MES-like model at the operating level is shown for item 2. The company level as located at item 3 is covered by an ERP-like process. Lastly, a workflow interface can be modelled at inter-company level depicted as item 4.

In the past, a major issue has been the definition of data interfaces. Through use of standards like EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport, ISO 9735:1988) a portion of these issues can be attenuated. However, this only solves the software part of the integration problem, i.e. the coupling of the information systems used in and between companies. Two further aspects complete the view:

- IoT (Internet of Things) technologies enhance the picture of a company that exists in its information systems by real time information.
- Process models extend the static view on a company by a dynamic one which allows to test future scenarios before they occur.

A model of a company which integrates data, processes, information systems and real time information concerning this company's resources is called its digital twin. It is a requirement to be able to develop cyber-physical systems for the industry of the future where the technical systems respond smartly to their environment.

The following section addresses the problems observed by the authors which limit the development of digital twins today. Research questions and methods derived from these observations are discussed in the third section. The main part of this paper elucidates the concepts of a Petri net-based environment to overcome these limitations. Further developments planned for the future are discussed at the very end.

CHALLENGES CONCERNING DIGITAL TWINS

A digital twin is a piece of software that models a part of the real – its real-world counterpart. For this, the digital twin stores all relevant status information of its real-world counterpart and is able to observe whether the real world behaves as predicted. Hence, the ability to predict the future is a central contribution of digital twins.

As a consequence, it is irrelevant whether this real-world counterpart actually exists or only its idea. For example, a production plant can already have a digital twin during its planning phase. Simulation can then be used to find out whether the planned plant will behave as assumed. All fabrication steps can be tested and optimized on the production line virtually (Kannwischer 2015). For this, the digital twin, when executed with real data, should behave like its real counterpart (Tao et al. 2017), or more precisely, it should calculate the same state as the one that can be observed in the real world.

Connectors from real world objects to their digital models are needed to compare the current and the predicted state and to deliver the relevant real-world information to the digital twins. Traditionally, sensors and user input have been these connectors. Hereby, sensors are typically restricted to local machines. User input, on the other hand, has often deficits concerning the correctness and currentness of data. In the past, these limitations have been accepted since they were the only available sources of information. In a business context a change can be observed nowadays as follows:

- Several information concerning business processes (like orders, bills, etc.) already primarily exist in the virtual world.
- By connecting machines and products to the internet, which is described by the term internet of things, also widely spread infrastructure information can be consolidated in a single place.

The other way around, actors controlled by digital twins might influence the reality. Again, the internet of things offers the ability to spread this information in a wide area.

Now, some of the major challenges are the differences concerning description languages for a business reality. They limit the ability to develop an integrated model. Two possible solutions exist for this:

- Like connectors in the real world it might be possible to link the different types of models with interfaces to each other.
- The different views on a business from the top floor to the shop floor are described in a monoparadigmatic way by usage of a small, generic language.

On first view, a lot seems to speak for the first solution since there exist several (business) process modeling languages that are widely used such as BPMN (Silver 2017), flowchart diagrams (information processing, ISO 5807:1985) or value stream diagrams (Klevers 2015). However, since no formal definition and semantics exists for these languages, the usefulness of interface definitions is hardly proofed. At field and production level, ladder diagrams are used as programming language when utilizing PLCs. Microcomputers (like a Raspberry Pi) are often coded on directly in a programming language such as C or C++. In each of these cases, the execution semantics are well defined, however no interface is defined concerning the above-mentioned diagrams yet.

As is illustrated in figure 2, this results in the goal of overcoming heterogeneous concepts between different business levels, while at the same time maintaining or – as a secondary effect in the case of data format mismatches – enabling the possibility to model systems integration beyond the company's borders. This can be achieved by the development of holistic digital twins.

Figure 2: Transition from Heterogenous Specification Languages to a Monoparadigmatic Description

In this paper, a monoparadigmatic approach is pursued based on Petri nets, which have been researched for more than 50 years and are clearly defined semantically (Reisig 2010). Petri nets allow for modelling on each of the levels mentioned in the later concept sections of this paper, from workflow management down to the controller – an advantage over other simulative languages that focus on specific applications like discrete event systems.

However, the existence of this method is not enough: in terms of applied sciences, also a web-based modeling and execution software is presented which has been developed over the past years and is currently applied to integration projects to overcome the limitations described above.

RESEARCH METHOD

According to (Hevner et al. 2004), there are seven guidelines for Design Science Research. These and their implementation are briefly explained as follows:

Design as an Artifact: A viable artifact must be created. In the present case, this is a web-based specification, simulation and control environment.

Problem Relevance: The goal is to develop a technology and/or methodology-based solution to a significant and pertinent problem. The presented artifact not only simulates business processes based on real data, but also optimizes and controls them.

Design Evaluation: The presented tool is used in companies already, thus feedback from the operational practice can be evaluated. The control part is used in teaching and prepared for operational use due to feedback from student users.

Research Contribution: The contribution consists of the creation of a practical application on the theoretical basis of the Petri net theory. The theory is enriched by time concepts and the possibility to connect Petri net models with the modelled reality.

Research Rigor: In theory, the simulation options developed are compared with other simulation environments. In practice the focus is placed on the evaluation of how to use this approach in a company environment.

Design as a Search Process: The present prototype is the latest in a series that starts from the initial implementation of the underlying principles and ends in a productive system. Each implementation step has been evaluated.

Communication of Research: The results achieved so far are relevant for both research and practice. They are presented on conferences but also, more illustrative, for practitioners.

CONCEPT AT FIELD LEVEL

Possibilities for using Petri nets for machine control and plant control have a long tradition (see (Zuse 1980), (König und Quäck 1988), (Hanisch 1992)). In addition to the specification and subsequent analysis of the possible behavior of a modeled system, Petri nets can also be used to generate code for PLCs (see (Brand 2000), (Raue 2001)).

Disadvantageous to such an approach is the need for a new code generation in case of adjustments to the machinery. This is not always useful in view of the desired flexible adaptability of machines due to altering customer demands.

An interesting and cost-effective variant is the use of microcomputers such as the Raspberry Pi to handle control tasks. If a web server is installed on the computer, then the web-based modeling and simulation environment presented here can also be installed. With its own data types and functions, complex process steps can be expressed compactly as a Petri net. By use of an extension allowing access to the GPIO interface of a Raspberry Pi, a production plant can be controlled directly by the simulator.

Starting point for such a control is the question as to how the actuators (for example motor or light) and sensors (for example button or light sensor) of the production line to be controlled are mapped in the tool and connected to it.

For actuators, it makes sense to represent them with a place that is linked to the appropriate GPIO interface by

means of an attribute. If the position is marked, then the actuator is switched on, otherwise switched off. If it is a continuously controllable actuator (for example a motor whose speed can be regulated), then the continuous variable is specified by the token value.

Sensors can also be represented by means of a place and in this case a token also indicates the condition of the sensor on that place, i.e. whether the sensor supplies a signal and, in the case of a continuous sensor, its value. However, in the latter case a sampling interval in which the sensor value is updated is also to be specified in expansion of the output interface. Furthermore, such places may only be read by transitions without deleting or modifying the token.

The example in figure 3 illustrates this principle as a Petri net using a production line with a processing station.
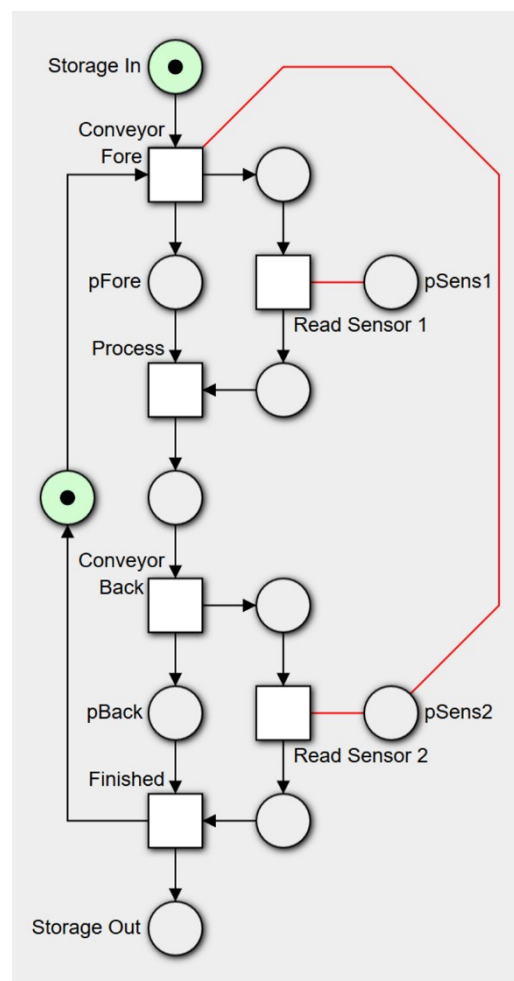

Figure 3: Process for Continuous Operation of a Two-Way Conveyor Belt with Attached Processing Station

Figure 4 shows the associated code specification except the part describing the layout has been left out to facilitate readability. In the initial state, the model waits for the signal from the light sensor at GPIO pin 24, which indicates that a work piece has arrived on the assembly line. As this is a discrete sensor, there is no need for interval sampling. When the signal arrives, the conveyor belt is set in motion in the direction of the workstation

until a signal at input port 23 is detected, thus indicating the correct positioning of the work piece. Now the processing will take place (which for reasons of space is not shown in detail here). Finally, the work piece is moved in the opposite direction back to the starting point.

```
N levelPLC {
  P pSem    (label='');
  P pSI     (label='Storage In');
  P pSO     (label='Storage Out');
  P pFore   (gpioOut=5);
  P pBack   (gpioOut=6);
  P pSens1  (gpioIn=23);
  P pSens2  (gpioIn=24);
  P p0, p1s1, p2s1, p1s2, p2s2 (label='');

  M (pSem=1);
  M (pSI=1);

  T tConvFore (label='Conveyor$Fore');
  T tReadSen1 (label='Read Sensor 1');
  T tPro      (label='Process');
  T tConvBack (label='Conveyor$Back');
  T tReadSen2 (label='Read Sensor 2');
  T tFin      (label='Finished');

  A (pSI, tConvFore, pFore, tPro, p0,
     tConvBack, pBack, tFin, pSO);
  A (tConvFore, p1s1, tReadSen1, p2s1, tPro);
  A (tReadSen1, pSens1);
  A (tConvBack, p1s2, tReadSen2, p2s2, tFin);
  A (tReadSen2, pSens2);
  A (tConvFore, pSens2);
  A (tFin, pSem, tConvFore);
}
```

Figure 4: Specification for the Model from Figure 3 (layout specification omitted for readability)

CONCEPT AT OPERATING LEVEL

Production planning focuses on the scheduling of orders and the aggregation of individual orders into lots. One way to solve this task has already been described in (Simon 2017) and (Simon 2018). Key is the ability to process several tokens put on one place in a sequence.
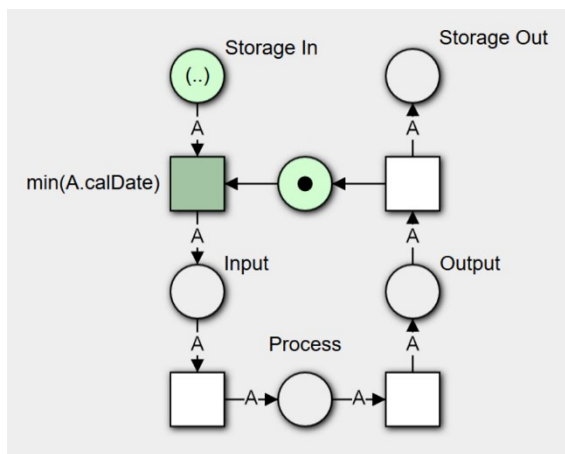

Figure 5: Process for Selection of Work Orders by Age

This corresponds to the possibility of sorting data records of a database with regard to a number of attributes in ascending and descending order. A comparable task can be seen in figure 5, where the oldest from a set of work orders is chosen by using a minimum selector. In this example, the orders planned at the company level are inserted into the production queue one by one. During production, the work pieces must first be fed to the workstation, then processed and finally taken away again.

The associated specification is depicted in figure 6 barring the part describing the layout.

```
N levelMES {
  R recOrder {calDate: datetime,
    var: char, amt: int};

  P pSem (label='');
  P p0 (label='Storage In', type=recOrder);
  P p1 (label='Input', type=recOrder);
  P p2 (label='Process', type=recOrder);
  P p3 (label='Output', type=recOrder);
  P p4 (label='Storage Out', type=recOrder);

  M (pSem=1);
  M (p0='("2018.04.02T08:00:00","oxo",75);
        ("2018.04.02T12:00:00","oxo",15);
        ("2018.04.10T08:00:00","oxo",25);
        ("2018.04.01T08:00:00","xo", 25);
        ("2018.04.10T12:00:00","xo", 50)');

  T t0 (label='', select='min(A.calDate)');
  T t1, t2, t3 (label='');

  A (p0, t0, p1, t1, p2, t2, p3, t3, p4,
     label='A');
  A (t3, pSem, t0);
}
```

Figure 6: Specification for the Model from Figure 5 (layout specification omitted for readability)

CONCEPT AT COMPANY LEVEL

From an information technology perspective, the company and the operating levels are quite similar: both levels process orders. But while at the plant level orders are brought into production, at the company level the business prerequisites to be able to process these orders are created.

The Architecture of Integrated Information Systems (ARIS) as shown in figure 7 is a coherent frame of reference for the various concepts used to describe a company from the point of view of information systems. The central role is played by the processes by which the data of a company is linked to its operational functions. Organizational responsibilities and the specification of relevant output data complete this approach.

Figure 7: Architecture of Integrated Information Systems (ARIS) as per (Scheer 1997)

Petri nets, which are mathematically defined, also offer the possibility of modelling business processes and are therefore used for this purpose, in particular for workflow

management systems (van der Aalst and van Hees 2002). They can be interpreted unambiguously and hence may serve as the basis for an automated workflow execution.

Distinctive tokens are crucial for the use of Petri nets to model and execute business processes. The best known approaches to this are Pr/T-nets (Predicate/Transition Nets) as per (Genrich and Lautenbach 1981) and Colored Petri Nets as per (Jensen 1992). The possible usage of the latter for modeling business processes is described exemplarily in (van der Aalst and Stahl 2011).

In Pr/T-Nets, places are typed and can be marked with matching tuples. Hence, places correspond to tables of a database and their tuples to the records within a database table. In this way, the network can be linked to a database or filled with real data via a CSV interface.

A sample application is a release process for orders in case all required materials are in stock. Otherwise, an order process will be triggered automatically. Figure 8 shows the model for this example using icons for visualization purposes.
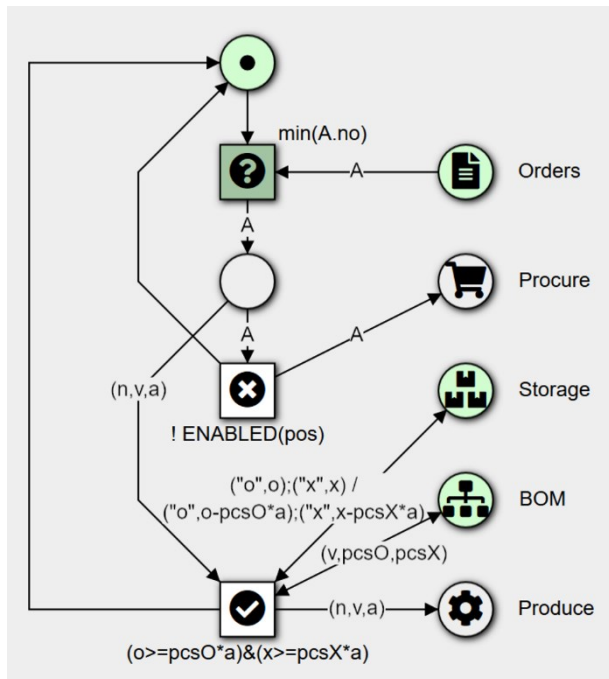

Figure 8: Process for Production Order Approval or for Repeat Order Placement, respectively

Figure 9 depicts the corresponding specification, again without mention of the layout portion.

```
N levelERP {
  R recOrder {no: int, var: char, amt: int};
  R recStorage {part: char, pcs: int};
  R recBOM {var: char, pcsO: int, pcsX: int};

  P p0 (label='');
  P p1 (label='', type=recOrder);

  P ord (label='Orders', type=recOrder);
  P sto (label='Storage', type=recStorage);
  P bom (label='BOM', type=recBOM);
```

```
  P proc (label='Procure', type=recOrder);
  P prod (label='Produce', type=recOrder);

  M (p0=1);
  M (ord='(1,"oxo",25);(2,"oxo",15);
          (3,"oxo",25);(4,"xo",25);
          (5,"xo",5)');
  M (sto='("o",200);("x",80)');
  M (bom='("oxo",2,1);("xo", 1,1)');

  T sel (label='', select='min(A.no)');
  T neg (label='', condition=
                '!ENABLED(pos)');
  T pos (label='', condition=
                '(o>=pcsO*a)&(x>=pcsX*a)');

  A (ord, sel, p1, neg, proc, label='A');
  A (p1, pos, prod, label='(n,v,a)');
  A (sto, pos, label='("o",o);
                     ("x",x)');
  A (pos, sto, label='("o",o-pcsO*a);
                     ("x",x-pcsX*a)');
  A (bom, pos, bom, label='(v,pcsO,pcsX)');
  A (neg, p0, sel);
  A (pos, p0);
}
```

Figure 9: Specification for the Model from Figure 8 (layout specification omitted for readability)

CONCEPT AT INTER-COMPANY LEVEL

The concepts at company and inter-company levels share innately even greater similarities than those at company and operating levels. If it is possible to model and simulate business processes on an ERP level, then connecting those with external processes on the same level becomes trivial.

The fundamental idea of this thought can be seen in the Workflow Management Coalition's Reference Model as depicted in figure 10. A company's workflows can be executed by workflow engines which themselves are parts of workflow enactment services. Through the use of appropriate interfaces, they may be connected to each other, even if they reside outside the company.
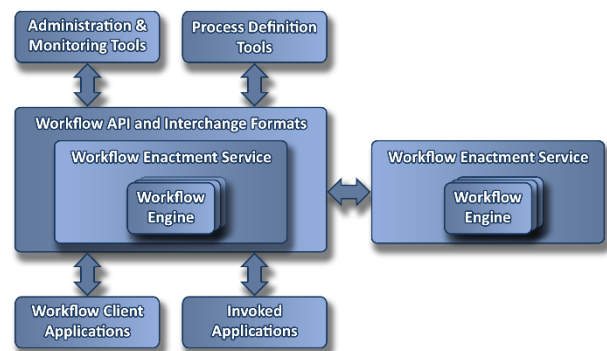

Figure 10: Components and Interfaces of the Workflow Reference Model as per (Hollingsworth 1995)

In connection with the possibility to use Petri nets as conceptual standard for the modelling and analysis of workflows (van der Aalst 1996), the claim made in the initial paragraph of this section proves true.

CONCLUSIONS

The integrated modeling and simulation environment for Petri nets presented in this article creates a monoparadigmatic and homogeneous approach for the simulation and control of processes at all levels of the automation pyramid. The system is already being used as part of consulting projects to optimize processes in retail and logistics, and currently production processes are being modeled, simulated and optimized on this basis in students' projects. Also the implementation of the system on a microcomputer is already used as part of lectures.

Overall, the results presented here make it clear that the goal of developing an integrated digital twin for both the various levels of the automation pyramid can be achieved with the help of Petri nets. A simulation of vertical integration from the top floor to the shop floor and back is already possible, as is one of horizontal integration along the supply chain.

FURTHER RESEARCH

In addition to the continuous expansion of the language base, new topics get queued due to the widespread use of the system, especially aspects of security. Different topics are relevant: The GPIOs of a Raspberry Pi control root privileges, which then have to be secured by the tool. However, before such topics can be addressed, the tool must be extended by a user management with access rights so that security-relevant information can be protected actually.

REFERENCES

van der Aalst. 1996. "Petri-net-based Workflow Management Software".In *Proceedings of the NFS Workshop on Workflow and Process Automation in Information Systems*, A. Sheth (Ed.). UGA, Athens, GA, 114-118.

van der Aalst, W. and K. van Hees. 2002. *Workflow Management – Models, Methods, and Systems*. MIT Press, Cambridge, MA.

van der Aalst, W. and C. Stahl. 2011. *Modelling Business Processes: A Petri Net-Oriented Approach*. MIT Press, Cambridge, MA.

Brand, A. 2000. *Generierung von SPS-Programmen aus Petri-Netzen*. Diplomarbeit, Universität Koblenz-Landau, Germany.

Genrich, H.J. and K. Lautenbach. 1981. "System Modelling with High-Level Petri Nets". *Theoretical Computer Science* 13, 1, 109-135.

Hanisch, H.-M. 1992. *Petri-Netze in der Verfahrenstechnik*. Oldenbourg, München, Germany.

Hevner, A.R.; S.T. March; J. Park; and S. Ram. 2004. "Design Science in Information Systems Research". *MIS Quarterly* 28, 1 (Mar), 75-105.

Hollingsworth, D. 1995. *Workflow Management Coalition: The Workflow Reference Model*. Workflow Management Coalition, Hampshire, United Kingdom.

International Organization for Standardization. 1985. *ISO 5807:1985 Information processing*. ISO, Genf, Switzerland.

International Organization for Standardization. 1988. *ISO 9735:1988 EDIFACT*. ISO, Genf, Switzerland.

Jensen, K. and L.M. Kristensen. 2009. *Coloured Petri Nets*. Springer, Berlin, Heidelberg, Germany.

Kannwischer, M. 2015. *Interaktive Präzisionswerkzeuge für die effizientere Bearbeitung. Produktivitätsfortschritte durch Industrie 4.0*. VDMA, Frankfurt, Germany (May).

Klevers, T. 2015. *Agile Prozesse mit Wertstrommanagement*. CETPM, Ansbach, Germany.

König, R. and L. Quäck. 1988. *Petri-Netze in der Steuerungstechnik*. VEB Verlag Technik, Berlin, Germany.

Raue, H. 2001. *Systematische Entwicklung einer Steuerung für eine Verpackungsmaschine*. Diplomarbeit, Universität Koblenz-Landau, Germany.

Reisig, W. 2010. *Petrinetze – Modellierungstechnik, Analysemethoden, Fallstudien*. Springer-Vieweg, Wiesbaden, Germany.

Scheer, A.-W. 1997. *Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse*. Springer, Berlin, Heidelberg, Germany.

Silver, B. 2017. *BPMN Quick and Easy Using Method and Style: Process Mapping Guidelines and Examples Using the Business Process Modeling Standard*. Cody-Cassidy, Los Angeles, CA.

Simon, C. 2017. "Eine Petri-Netz-Programmiersprache und Anwendungen in der Produktion". In *AKWI Tagungsband 2017*, T. Barton; F. Herrmann; V.G. Meister; C. Müller; and C. Seel (Eds.). mana-Buch, Heide, Germany, 61-70.

Simon, C. 2018. "Web-based Simulation of Production Schedules with High-level Petri Nets". In *Proceedings of the 32nd European Conference on Modeling and Simulation* (Wilhelmshaven, Germany, May 22-25), 275-281.

Tao, F.; J. Cheng; Q. Qi; M. Zhang; H. Zhang; and F. Sui. 2018. "Digital Twin-Driven Product Design, Manufacturing and Service with Big Data". *The International Journal of Advanced Manufacturing Technology* 94, 9-12 (Feb), 3563-3576.

Wolf, T. and J.-H. Strohschen. 2018. "Digitalisierung: Definition und Reife". In *Informatik-Spektrum* 41, 1 (Feb), 56-64.

Zuse, K. 1980. *Petri-Netze aus der Sicht des Ingenieurs*. Springer-Vieweg, Wiesbaden, Germany.

AUTHOR BIOGRAPHIES

STEFAN HAAG holds degrees in Business Administration and Engineering as well as Economics with his main interests being related to modelling and simulation in graphs. After working at the Fraunhofer Institute for Systems and Innovation Research ISI Karlsruhe for several years, he is now a Research Fellow at the Hochschule Worms. His e-mail address is: `haag@hs-worms.de`.

CARLO SIMON studied Informatics and Information Systems at the University of Koblenz-Landau. For his PhD-Thesis, he applied process thinking to automation technology in the chemical industry. For his state doctorate, he considered electronic negotiations from a process perspective. Since 2007, he is a Professor for Information Systems, first at the Provadis School of Technology and Management Frankfurt and since 2015 at the Hochschule Worms. His e-mail address is: `simon@hs-worms.de`.