Research Center
Finance & Information Management

Fraunhofer FIT
Project Group
Business & Information
Systems Engineering

Research paper

Modelling Availability Risks of IT Threats in Smart Factory Networks – A Modular Petri Net Approach

by

Stephan Berger, Maximilian Bogenreuther, Björn Häckel, Oliver Niesel

March 2019

University of Augsburg, D-86135 Augsburg
Visitors: Universitätsstr. 12, 86159 Augsburg
Phone: +49 821 598-4801 (Fax: -4899)

University of Bayreuth, D-95440 Bayreuth
Visitors: Wittelsbacherring 10, 95444 Bayreuth
Phone: +49 921 55-4710 (Fax: -844710)

www.fim-rc.de

WI-949

MODELLING AVAILABILITY RISKS OF IT THREATS IN SMART FACTORY NETWORKS – A MODULAR PETRI NET APPROACH

Research paper

Abstract

In manufacturing, concepts like the Internet of Things (IoT) or Cyber-physical Systems (CPS) accelerate the development from traditional production facilities towards smart factories. Thereby, emerging digital technologies increasingly connect information networks with production processes, forming complex smart factory networks. Due to their reliance on information flows and the high degree of cross-linking, these networks are, in particular, vulnerable to availability risks caused by attacks and errors. To address this problem, we aim to identify and analyse availability threats by developing a modelling approach that depicts specific characteristics of smart factory networks. Based on modelling requirements derived from a literature review, we propose a modular Petri net approach. To iteratively revise and validate our model, we followed established evaluation principles and conducted evaluation rounds with industry experts and other researchers. To demonstrate the usefulness and applicability of our model, we simulated one real-world use case and two planned extensions of a mechanical engineering company. Our model depicts information-based dependencies within smart factory networks and allows for the simulation and analysis of threat propagation. Thereby, it enables both researchers and practitioners to identify critical network connections and components, serving as a basis for layout decisions and IT security mitigation measures.

Keywords: Smart Factory Network, Information Network, Production Network, Availability Risks, Attack Propagation, Petri Nets.

1 Introduction

In manufacturing, concepts like the Internet of Things (IoT) and Cyber-physical Systems (CPS) accelerate the development from traditional production facilities towards smart factories (Lasi et al., 2014). These self-organised, autonomous systems increase flexibility and production efficiency by monitoring and controlling production processes in real-time (Brettel et al., 2014; Lasi et al., 2014; Radziwon et al., 2014). Within smart factories, the widespread use of digital technologies fosters the fusion of information networks with the production processes towards complex smart factory networks (SFN) (Häckel et al., 2018). Thereby, the increasing availability and exchange of data enables the flexible production of customised products down to lot size one, while simultaneously improving production efficiency (Radziwon et al., 2014). However, due to the high cross-linking of IT systems and physical production components, SFNs become ever-more complex, which rises the probability of disturbances and errors (Broy et al., 2012; Tupa et al., 2017). Furthermore, the increasing openness and integration into their socio-economic environment make SFNs, in particular, vulnerable to IT security risks (Smith et al., 2007; Yoon et al., 2012; Tupa et al., 2017; Häckel et al., 2018).

Besides integrity and confidentiality, especially availability threats affect SFNs (Amiri et al., 2014). As physical production processes increasingly rely on information of underlying IT systems, availability threats affect both production and information networks (Broy et al., 2012). Threat propagation further amplifies the vulnerability within highly interconnected networks (Smith et al., 2007), possibly causing an entire network to break down (Amiri et al., 2014). A study among 900 companies revealed that 70% have been target of IT attacks, whereby 50% of successful attacks led to downtimes within the production process or loss of operations (BSI, 2017). Apart from intentional IT attacks, unintentional errors may also result in discontinuance or limitation of production (Amiri et al., 2014). Interconnectedness thereby promotes threat propagation, while production concepts like just-in-time further increase potential damage. Hence, organisations face the challenge of foreseeing and analysing availability threats caused by IT attacks and errors within SFNs. To increase transparency and enable risk management, an adequate modelling approach is required which is able to capture SFN components and dependencies as well as impacts of availability threats. Thus, the identification of critical network entities becomes feasible and serves as a foundation for SFN layout decisions and IT security mitigation measures. Till date, existing works either focus on selected parts of the issue, e.g. modelling network entities (e.g. Zhu et al., 2011; Darwish and Hassanien, 2017), IT availability risks (e.g. Rainer et al., 1991; Häckel et al., 2018), or use insufficient methods to depict and simulate both SFNs and/or threat propagation (e.g. Kuo and Huang, 2000; Selic and Gérard, 2013). To the best of our knowledge, a holistic, integrated approach combining a suitable modelling approach, depiction of SFNs, and threat propagation analysis is missing. Against this backdrop, we formulate the following research question:

How can availability risks of IT attacks and errors in smart factory networks be modelled?

To answer our research question, we conducted a literature review and derived formal and functional modelling requirements for our model. We then followed the design and evaluation guidelines of Sonnenberg and Vom Brocke (2012) to iteratively develop our modelling approach, which is based on classical Petri Nets (PN) (Petri, 1966) and multiple PN extensions. Besides enabling visual depiction and mathematical analysis, our modular PN approach exhibits specific smart factory properties and behaviour. Based on the depiction of information and production components and their dependencies, we perform detailed simulation and analysis of attack and error propagation to examine availability threats and associated network effects. With this, we create transparency within complex SFNs and support decision making concerning SFN layouts, e.g. by comparing the availability of single components for different setups or integrating redundant information components.

The remainder of this paper is organised as follows. In Section 2, we elaborate on availability threats in SFNs and discuss existing works. From this, we derive requirements for a modelling approach and outline our research method based on PNs in Section 3. In Section 4, we present our modelling approach as the core of our work. For evaluation, we simulate a real-world use case with two planned extensions to demonstrate the applicability of our model and present feedback of focus groups and industry experts which supplemented our model in Section 5. Section 6 summarises and concludes our work.

2 Theoretical Background

In the following, we present related work to underpin our research objective. More precisely, we review the current body of literature on SFNs and associated availability risks. Based on literature, we derive general and SFN-specific modelling requirements for our development process within the next section.

Although the term *smart factory* is widely used in research and practice, a common definition is missing. Summarising literature on SFNs, Radziwon et al. (2014) generally define smart factories as systems which solve production problems in a highly dynamic environment. Increasingly merging the cyber and physical world, concepts like IoT and CPS integrate smart objects such as smart machines and products into smart factories (Lucke et al., 2008; Lee et al., 2015; Wang et al., 2016; Zhang et al., 2016). "Within modular structured smart factories […], CPSs monitor physical processes, create a virtual copy of the physical world and make decentralised decisions" (Hermann et al., 2016). The cross-linking of CPS entities forms decentralised, functional divisions of information supply and processing (Brettel et al., 2014). In SFNs, divisions with similar functions, e.g. embedded systems, are grouped to layers, building an information hierarchy (Lee et al., 2015), which is needed to coordinate decentralised entities (Schuh et al., 2014). Decentralisation and modularisation lead to flexibility, configurability, and adaptability (Zuehlke, 2010; Brettel et al., 2014). The optimisation of processes, paired with further benefits like resource efficiency and ever-shorter lead times, enhances productivity (Brettel et al., 2014; Schuh et al., 2014) and simultaneously enables the production of customised products down to lot size one in ever-shorter time-to-market and at competitive costs (Lasi et al., 2014).

Due to increasing information use and complex interdependencies within production processes, SFNs are, in particular, vulnerable to disruptions within underlying information systems (Pasqualetti et al., 2013). Disruptions or threats are caused by intentional attacks or unintentional errors. Thereby, the source of an attack is internal, e.g. social engineering by employees, or external, e.g. Denial-of-Service attack by cybercriminals (Cardenas et al., 2009). Errors are of technical, operational, or organisational origin (Amin et al., 2013). To prevent and mitigate damage within information intensive production, the CIA triad, i.e. confidentiality, integrity, and availability, provides guidance for IT security within organisations (Cardenas et al., 2008; Sadeghi et al., 2015). Integrity describes the trustworthiness of the information, while confidentiality aims for keeping critical data secret (N. W. Group, 2008). Availability refers to the functioning of components. The high degree of interconnectivity makes SFNs particularly vulnerable to availability risks in both information hierarchy and production capability (Broy et al., 2012; Cardenas et al., 2008). Moreover, the cross-linking between modules within the SFN hierarchy favours cascading effects of attacks and errors, eventually causing the breakdown of the entire system (Danziger et al., 2016). Real-time interactions between information and production networks further leverage effects of availability losses (Amiri et al., 2014). With the increasing reliance on the information network and real-time requirements, availability threats therefore become one of the most critical threats within SFNs (Cardenas et al., 2008; Lee, 2008; Amiri et al., 2014).

In literature, multiple works discuss the depiction of complex networks and the assessment of availability risks by using various modelling approaches. Ivanov et al. (2016) use *mathematical notation* to model short-term supply chain scheduling in dynamic systems. Focusing on the optimisation of production in SFNs on a technical level, however, the approach does not depict the propagation of threats. In addition, general models should be comprehensive for all stakeholders (Selic and Gérard, 2013, p. 4), which is not ensured in a purely mathematical form. Selic and Gérard (2013) use an extension of the *unified modelling language* (UML) called modelling and analysis of real-time embedded systems (MARTE) to model CPSs. Being used for software engineering and supporting the transfer into software code, UML provides a graphical representation of complex structures. MARTE adds quantitative and qualitative measures, timing, and other CPS features. However, this modelling approach lacks a dynamic illustration of events such as threat propagation. Based on *system dynamics*, i.e. a method for analysing and simulating dynamic, complex systems, Genge et al. (2015) introduce Cyber Attack Impact Assessment (CAIA). CAIA focuses on cascading effects of attacks and tracks the functionality of a system by monitoring critical changes in production. However, this approach is insufficient regarding the depiction of hierarchical dependencies between IT layers. Various approaches

use *graph theory* to depict and measure risks in supply chain networks and critical infrastructures (Wagner and Neshat, 2010; Buldyrev et al., 2010), however, neglect the depiction of dynamic propagation effects. Wu et al. (2007) use *PNs* to model propagation effects in supply chains. Considering dynamic and stochastic behaviour, Fridgen et al. (2015) introduce a modular PN approach to simulate and analyse exogenous shocks in supply networks. Hence, PNs are appropriate to model and analyse hierarchical structures (Salfner and Wolter, 2009) as they provide graphical notation, formal language (Billington, 1988; van der Aalst, 1993), and support the modelling of dynamic and stochastic behaviour (Arns et al., 2002). Moreover, modular PNs proved to be suitable to enhance transparency in complex SFNs (Tsinarakis et al., 2003; Fridgen et al., 2015).

3 Research Method

For our approach, PNs are the appropriate method to investigate availability risks in SFNs. Originally developed by Carl Adam Petri (Petri, 1966), soon the methods' suitability and usefulness for practical purposes showed the need for more advanced PN models (Jensen, 1987). Ever since, PNs continuously enhanced and multiple powerful extensions emerged. Today, PNs are widely used to describe all kinds of industrial applications like shocks in complex supply networks (Fridgen et al., 2015), flexible manufacturing systems (van Brussel et al., 1993; Venkatesh and Zhou, 2000), and automated production networks (Boucher et al., 1989; D'Souza and Khator, 1994; Long et al., 2016). Thereby, PNs enable the depiction, simulation, and analysis of network structures and layouts (Fridgen et al., 2015). In addition, the method allows for modelling internal and external events and their effects on the overall system (Razzaq and Ahmad, 2015; Szpyrka and Jasiul, 2017). Meeting the urgent need for transparency, PNs are suitable for depicting SFNs and simulating attacks, errors and their propagation.

The general concept of PNs is based on bipartite graphs, which consist of places (p_n), transitions (t_n), and arcs (Figure 1). Arcs connect places and transitions, whereby a transition has at least one input (p_1, p_2) and/or output place (p_3). Places may contain a discrete number of 'tokens'. The number of tokens per place is the 'marking' of the system. The marking changes, i.e. tokens move, when a transition is enabled and 'fires'. A transition is enabled when all input places contain the number of required tokens.

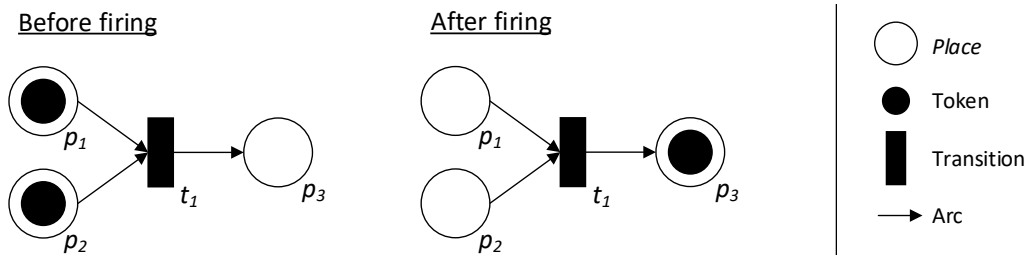


Figure 1. *Petri Net Example*

Based on the literature review in Section 2, we derived formal and functional modelling requirements to depict availability risks in SFNs. In the following, we outline how classical PNs (Petri, 1966) and their extensions fulfil these requirements.

First, we discuss formal requirements covered by classical PNs (Table 1). The major challenge of SFNs is to overcome the growing complexity, which continuously increases through ever growing cross-linking and multi-functionality of systems. To foster transparency and clarity, PNs not only provide a *graphical representation* (FoR.1), but also offer the possibility to transfer the depicted network into a *mathematical notation* (FoR.2) to enable further analysis (van der Aalst, 1998). Moreover, we demand for *scalability* (FoR.3) to facilitate the adaption of our model and to reduce opacity of large network structures. We therefore use the concept of modularisation (Dotoli and Fanti, 2005). By considering a network as a composition of multiple information and production modules, we allow for both simple extendibility and analysis of selected network parts, i.e. single modules, as well as the overall network.

	#	Requirement	Purpose	Method
Formal	FoR.1	Graphical representation	Depiction in a visual, comprehensive form.	Petri Nets
	FoR.2	Mathematical notation	Analysis in terms of simulation or indicator-based comparison of different network layouts.	Petri Nets
	FoR.3	Scalability	Analysis of network parts and simple extendibility through modularisation.	Petri Nets

Table 1. Formal Modelling Requirements for Smart Factory Networks

Second, we present functional requirements which, in particular, account for specific properties of SFNs (Table 2). As elements of classical PNs are insufficient for modelling SFNs, we used multiple PN extensions. For our approach, we define a SFN as the combination of an information network and a production network in a smart factory without considering interconnections within its socio-economic environment, i.e. suppliers or customers. The information and production network include multiple modules. To enable the modelling of an IT layer hierarchy, e.g. from a central server down to embedded systems, and to link information and production network, we require *inter-modular dependencies* (FuR.1). IT layers include modules with similar functionality, whereby modules on the same IT layer are only connected to modules on adjacent layers. To depict these dependencies, we applied guard functions introduced by 'Coloured Petri Nets' (CPN) (Jensen, 1991). A guard function maps a Boolean value to transitions, indicating whether the recent marking enables the transition or not. Thereby, the markings on places from other modules are checked. For our approach, we need guard functions to depict modules on different IT layers and simultaneously obtain the requirement of *scalability* (FoR.3), i.e. maintaining separate modules which are connected to modules on adjacent layers. Furthermore, we account for *temporal behaviour* (FuR.2) by using 'Timed Coloured Petri Nets' (TCPN) to enable the depiction of asynchronous, time-delayed process steps, e.g. production or repair time (Ramchandani, 1973). TCPNs introduce a global time and a token timestamp. The global time defines the time of the system, the token timestamp, i.e. enabling time (van der Aalst, 1993), defines the point in global time a token can be used. Within the information network, we demand for *stochastic behaviour* (FuR.3) to model dynamic attack and error propagation as well as stochastic error and attack duration. We, therefore, introduce 'General Stochastic Petri Nets' (GSPN), which focus on distributed firing times of transitions. Within the production network, *product customisation* (FuR.4) requires the assignment of individual characteristics to products, e.g. to model lot size one. Again, we use CPNs which assign colour sets, i.e. information, to tokens. Each place accepts only tokens of a predefined colour set (Jensen, 1987). Finally, we require *machine parallelisation* (FuR.5) to cover various production layouts. To align with our modularisation concept and reduce complexity, we use *redundant place nodes* (De La Mota et al., 2017). Facilitating the development and comprehension of models, redundant place nodes are duplicate nodes with the same name and number of tokens (De La Mota et al., 2017). Therefore, the copied place and its duplication can be seen as one.

		#	Requirement	Purpose	Method
Functional	**Smart Factory Network**	FuR.1	Inter-modular Dependency	Modelling of IT layer hierarchy and link of information and production network.	Coloured Petri Nets
		FuR.2	Temporal Behaviour	Modelling of state durations to enable asynchronous processes.	Timed Coloured Petri Nets
	Information Network	FuR.3	Stochastic Behaviour	Modelling of stochastic attack and error propagation through stochastic events.	General Stochastic Petri Nets
	Production Network	FuR.4	Product Customisation	Modelling of various product characteristics, e.g. to model lot size one.	Coloured Petri Nets
		FuR.5	Machine Parallelisation	Modelling of various production layouts.	Redundant Place Nodes

Table 2. Functional Modelling Requirements for Smart Factory Networks

Besides fulfilling formal and functional modelling requirements, the presented PN approaches implicitly offer additional design options for SFNs. First, PNs allow to capture states of modules in discrete time. We use this to distinguish the availability status of an information module, i.e. operational, on hold, failed after attack, and failed after error. Second, different arc types reduce complexity and foster compact models (Verbeek et al., 2010). We introduce reset, test, and inhibitor arcs in Section 4. Third, PNs enable the modelling of simple and smart production machines (PM). In contrast to a simple PM, a smart PM requires product-specific information for product customisation (FuR.4).

We gradually revised and evaluated our PN model in the course of three iterations (Figure 2). Aiming for a structured development process, we applied the three evaluation principles of Sonnenberg and Vom Brocke (2012): (1) *distinction between internal and external actions* for developing the model, (2) *documentation* of design theories, and (3) *continuous assessment* of the development process. To do so, we developed and evaluated our model with both researchers and practitioners. We supplement the results of our internal modelling activities with feedback from two focus group discussions with other researchers and an industry expert interview. In accordance with the evaluation principles, we worked out semi-structured questions for both external evaluation forms to guide the discussions without limiting the creativity of the participants' feedback. Principle (1) aims to evaluate design decisions and ensures the usefulness of the model. Hence, we presented the initial version of our modelling approach to our focus group and discussed its structure and our modelling requirements. Being potential users of our model, we repeated this procedure with our industry experts and closed the feedback loop by consulting the same focus group within a third iteration. To comply with principle (2), we ensured structured documentation by meeting Gregor and Jones's (2007) eight elements of design theory, i.e. scope, constructs, principles of form and function, principles of implementation, artefact mutability, testable propositions, justificatory knowledge, and expository instantiation, which we discuss in Section 5. Regarding principle (3), we iteratively revised our model and defined the following ending conditions for our development process: fulfilment of formal and functional modelling requirements (objective condition) and agreement of relevance, applicability, and usability of the model (subjective condition). To offset potential bias, we checked the subjective ending condition in cooperation with our focus group and industry experts. At the end of each iteration we checked whether the ending conditions were met. After the third iteration entailed only marginal adjustments and the focus group confirmed the validity of our model, we refrained from conducting another iteration and completed the development process.

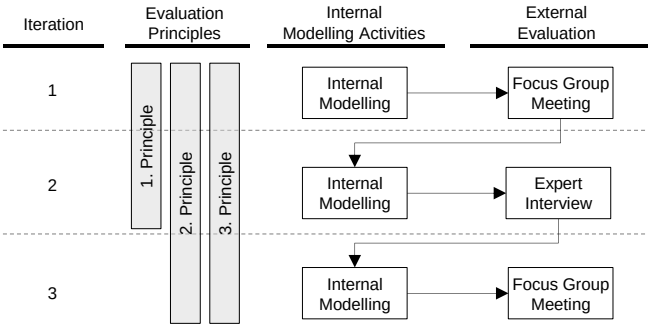


Figure 2. Iterative Development Process of our Modelling Approach

Focus group meetings enable open discussions in which the participants take turns in alternately arguing and thereby build on each other's opinions (Krueger and Casey, 2014). Besides using feedback within the development process, this method is further suitable to evaluate and demonstrate the utility of a model (Tremblay and Berndt, 2010). The focus group meetings were hosted by all four co-authors and lasted 90 minutes each. Our focus group encompassed eleven members of two different universities and consisted of one professor, eight doctoral students, and two Master students. All members have an IS background and focus on digitalisation topics such as SFNs or IT security. Expert interviews provide "insights into or understanding of opinions, attitudes, experiences, processes, behaviours, or prediction" (Rowley, 2012, p. 261). Feedback from practice is especially valuable as the industry experts are intended users of the model. We conducted our interview with two experts of a globally operating manufacturer in the mechanical engineering sector with 6,600 employees and an annual revenue of EUR

1.3 billion. The first expert is a presales manager with a focus on digitalisation topics, while the second interviewee deals with big data concepts within SFNs. The interview with both experts took three hours.

To evaluate the applicability and usefulness of our model, we created and parameterised three scenarios, i.e. a real-world use case and two planned extensions, with the industry experts. We then simulated those scenarios, calculated selected key performance indicators, and discussed the results with the experts. Our scenarios demonstrate how our model supports design decisions for different SFN layouts.

4 Petri Net Model for Smart Factory Networks

In this section, we introduce our approach for depicting availability risks within SFNs (Figure 3). In our model, SFNs consist of an information network including an IT layer hierarchy and a production network. We modelled both networks separately to simulate the propagation of attacks and errors within the information network and show the effect on the production network. Thereby, the information network unidirectionally provides product-specific information to the production network, illustrating the information flow for product customisation (FuR.4). Within the production network, simple and smart PMs process products.

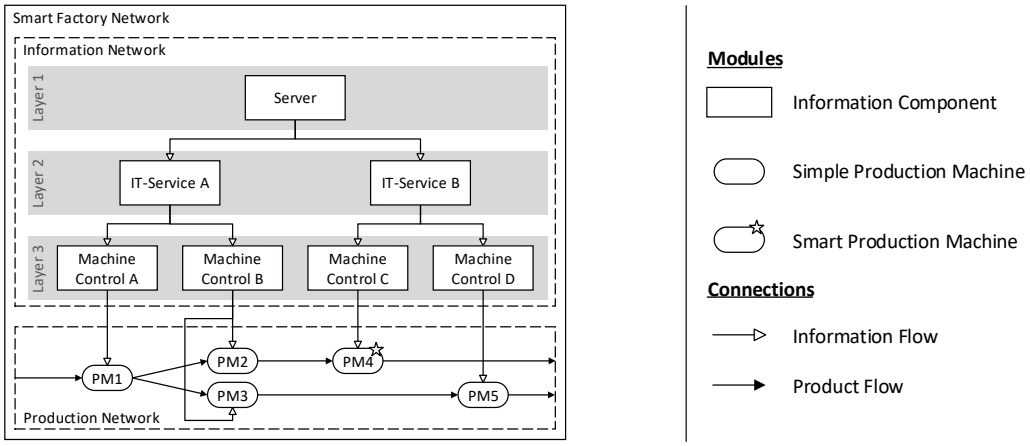


Figure 3. Exemplary Layout of a Smart Factory Network

4.1 Information Network

In literature, various approaches for modelling information networks exist (Thomesse, 2005; Zuehlke, 2010; Interntaional Society of Automation, 2010). These approaches mainly differ in the number of included (IT) layers. Thereby, each layer fulfils a specific functionality such as storage, transmission, manipulation, or processing of information and includes at least one Information Component (IC). To fulfil the requirement of scalability (FoR.3), we depict ICs as identically structured modules, which enables the modelling of any number of layers. The information network in Figure 3 consists of three layers: the server layer, the IT-service layer, and the machine control layer. Except for ICs on the highest hierarchic level, all ICs are connected to at least one IC on a higher layer. Redundancies can be depicted by connecting ICs to more than one IC on a higher layer. Due to hierarchic modelling, failure of an IC cascades down to connected ICs on lower layers.

To enable profound analysis of availability risks, i.e. attack and error propagation, we designed ICs with four states (Figure 4): *operational* (OP), the semi-functional state *on hold* (OH), and the non-functional states *failed after error* (FE) and *failed after attack* (FA) (Häckel et al., 2017). Each state is represented by a place. Starting from the OP state, two transitions are associated with each of the semi- and non-functional states: one transmission allows for entering the respective state, and one for exiting it. Interfaces (i_n) represent dependencies between ICs within the information network. Interfaces are not elements of PNs, however, they support the visualisation of guard functions attached to transitions t_1, t_2, and t_5 handling information from connected ICs. Thereby, interface i_1 provides information about the state of connected ICs, whereas i_2 serves as a trigger for the propagation of an attack. Tokens of ICs are

not coloured as they solely contain a timestamp. In addition, tokens stay within an IC, as ICs only consist of intra-modular arcs. Besides regular arcs, we use *reset arcs* consuming all tokens from attached places (Christensen and Hansen, 1993; Dufourd et al., 1998) and *test arcs*, which do not consume tokens and therefore cannot change the marking of a place (Christensen and Hansen, 1993).

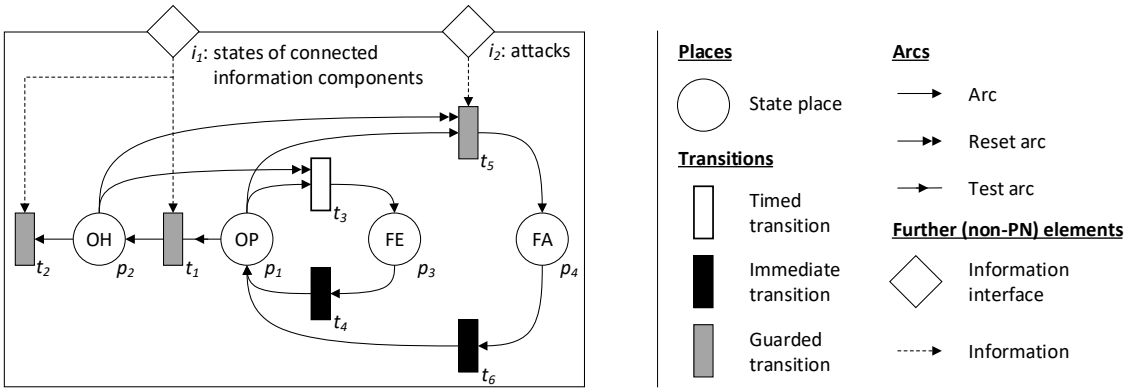


Figure 4. Structure of an Information Component

In the following, we describe the functioning of the defined IC states in more detail (Figure 5). *Operational* is the default state of an IC, meaning that the component is fully functional and provides information to underlying components. An IC is able to enter the states OH, FE, and FA through transitions t_1, t_3, and t_5. As transitions may be enabled simultaneously, e.g. occurrence of error and attack at the same time, we implemented a prioritisation rule from t_5 to t_1, with t_5 holding the highest priority.

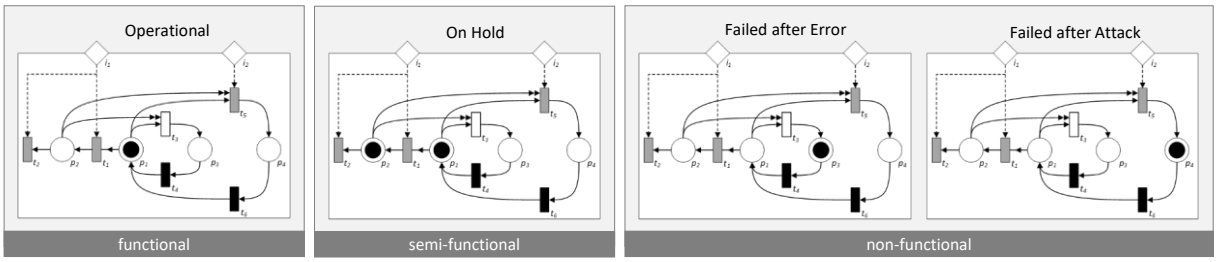


Figure 5. States of an Information Component

To capture interruptions in the information flow, we implemented the semi-functional state *on hold*. An IC is OH when it cannot receive information from connected ICs on higher IT layers, i.e. these ICs are not OP. Therefore, interface i_1 provides the current states of connected ICs to the guard functions of transitions t_1 and t_2. To depict the semi-functional state of an IC, we used a test arc to model the input arc of t_1. Thus, the token remains on p_1 while another token is put onto p_2. As, in our model, multiple connections to ICs on a higher layer depict redundancies, the token on p_2 is removed, i.e. the IC returns to OP, when at least one connected IC on the layer above returns to OP. p_2 is also reset when the IC enters FE or FA. We explain the effect of an IC in OH on the production network in Section 4.3.

Failed after error represents a non-functional IC with an error. The time until an error occurs follows an exponential distribution with an IC-specific error rate λ_E. When a token is placed on p_1, the timed transition t_3 receives a random time. The transition fires as soon as the global time equals the assigned time. Since an IC can enter FE from both OP and OH, p_2 is reset when t_3 fires. Then, the only token is on p_3 and the IC is FE. The duration of an error follows an exponential distribution with an IC-specific error recovery rate λ_{ER}. The error duration is added to the current global time and set as timestamp for the token. When the token is activated, t_4 fires immediately and the IC returns to OP. Assuming that an erroneous IC will be disconnected from the network and repaired immediately, it cannot be attacked as long as it stays in FE.

After a successful attack, an IC enters the *failed after attack* state. Thereby, the IC is either the primary target of an attack or subject to an attack propagation. In the first case, the IC will enter FA unless it is FE. For the second case, we implemented a propagation logic which is based on the assumption, that

the security gap of an attacked IC is closed by the time the IC has recovered. Thus, an attack affects an IC only once. If an attack, that an IC has not suffered from already, is anywhere within the network, i.e. at least one IC is FA (depicted by interface i_2), the guard function of transition t_5 returns true. To avoid immediate infection of the entire information network, however, the guard function of t_5 only returns true at an IC-specific propagation rate λ_P. When these conditions are fulfilled, t_5 fires and the IC enters the FA state. The attack duration follows an exponential distribution with an IC-specific attack recovery rate λ_{AR} and is influenced by the severity of the attack. The value obtained is added to the current global time and set as timestamp for the token. When global time matches the timestamp, the token is activated, t_6 is enabled and fires immediately, and the IC returns to OP.

4.2 Production Network

Analogous to the information network, we chose a modular structure for modelling scalable production networks. For depicting different production layouts, we arrange our PMs either sequentially or in parallel. Thereby, each PM conducts one production step, e.g. painting the product. One production step, however, can be conducted by multiple PMs. Therefore, the production network does not represent the logical sequence of production steps, but the physical flow of products.

A PM includes two storage places (s_n), two transitions, and one regular place (Figure 6). Storage places represent temporary depositories between PMs. To account for different levels of smartness, we introduce two types of PMs: (1) smart PMs requiring product-specific information to process customised products, and (2) simple PMs, which treat every product equally without using additional information. Thus, the guard function of the PM uses information about the states of connected ICs provided by the interface i_1. Besides, information about the required order and type of production steps is needed for product customisation. Thus, we used coloured product tokens to depict production information, e.g. a required production step. Furthermore, we used *inhibitor arcs* to illustrate that a PM processes only one product at a time. This arc type is often used to describe pure sequences and restricts a transition from being enabled if a connected place is occupied (Peterson, 1977; Janicki and Koutny, 1995).

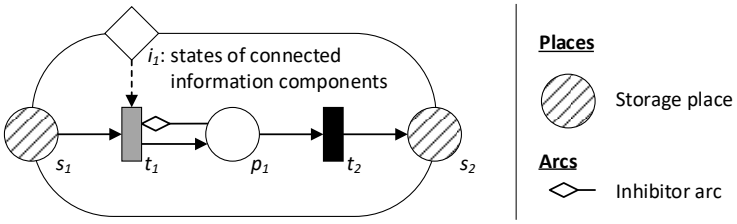


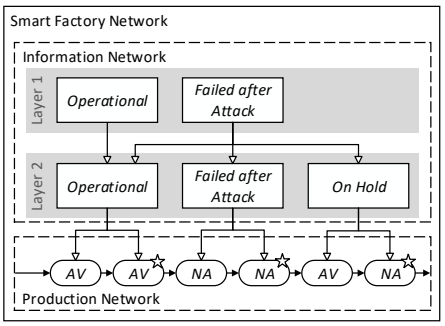
Figure 6. Structure of a Production Machine

Except for the guard function, smart and simple PMs are identical. Within the production network, tokens represent physical products. In contrast to the information network, product tokens are not bound to one PM but get processed by multiple PMs. Only tokens with an adequate colour, i.e. the required production step, can enter a PM. A token on place p_1 is being processed. Since PMs can only process one token at a time, the inhibitor arc prevents transition t_1 from firing as long as p_1 is occupied. When a token enters p_1, the token receives a token timestamp equalling the duration of the production step to be performed. Once global time matches the timestamp, t_2 fires immediately and transfers the token to (output) storage place s_2. We use redundant place nodes to model the transit of products between PMs. Thereby, the output storage place of a PM equals the input storage place of all subsequent PMs.

4.3 Smart Factory Network

A SFN merges information and production network. The interconnection between an IC on the lowest IT layer and a PM is 1:n, i.e. each PM is assigned to exactly one IC, while ICs may be connected to several PMs. Due to unidirectional information flow, the state of a PM depends on the state of the connected IC. Smart and simple PMs are *available* (AV) when the connected IC is functional, and *not available* (NA) when it is non-functional (Figure 7). The OH state represents a functional IC, implying

connected ICs on higher layers to be *non-functional*. Consequently, ICs OH do not provide information to smart PMs. In case a connected IC is OH, smart PMs are NA, while simple PMs remain AV.



	State of Connected Information Component			
	Operational	**On Hold**	**Failed after Error**	**Failed after Attack**
Simple Production Machine	AV	AV	NA	NA
Smart Production Machine	AV	NA	NA	NA

AV = available; NA = not available

Figure 7. States of Production Machines depending on Information Components

To enhance production efficiency, we implemented a reallocation logic which redirects products waiting for NA PMs. The duration of the reallocation, i.e. transfer time of waiting products to other PMs, depends on the layout of the production network. A waiting product is reallocated if the considered PM fulfils all following conditions: (1) the PM conducts the required production step, (2) the PM is available, and (3) the PM has the lowest reallocation time of all PMs fulfilling the aforementioned conditions.

5 Evaluation and Application

Aligning with the three evaluation principles of Sonnenberg and Vom Brocke (2012), we ensured *distinction between internal and external actions* (see Section 3). The *documentation* of our prescriptive design theories is based on the eight design theory elements of Gregor and Jones (2007), comprising scope, i.e. research question, constructs, i.e. current version of our model, principles of form and function, i.e. FoR.1-3, and principles of implementation, i.e. FuR.1-5. We integrated the remaining four design elements, i.e. artefact mutability, testable propositions, justificatory knowledge, expository instantiation, within the collected feedback of our focus group meetings and expert interview (see Section 5.2). Accounting for *continuous assessment*, we revised our model under consideration of our formal and function modelling requirements and feedback from our external evaluations.

In the following, we demonstrate the practical application of our model: first, we illustrate a real-world use case and two planned extensions. Second, we provide detailed feedback of the focus group meetings and expert interviews, including associated adaptions of the model.

5.1 Real-world Use Case and Planned Extensions

MILLING is one of the world's leading manufacturers of customised milling and turning machines as well as auxiliary products that facilitate their use. For our real-world use case, we depict the process of manufacturing metal pallets for the automotive industry. A metal pallet is used to attach one or several products to be manufactured, e.g. engine blocks or chassis, and therefore serves as an auxiliary construction for transportation. To attach a product to a pallet, the pallet is equipped with holes and T-slots. As the attached products vary in size and shape, pallets are suitable for customisation concerning the number, position, and size of holes and T-slots. As the production is also highly automated, MILLING is an intended user of our model and suitable for practical evaluation.

To demonstrate the suitability of our model to support design decisions for SFNs regarding availability risks, we created and simulated three application scenarios in cooperation with MILLING (Figure 8). As basic *scenario 1*, we depict the status quo of pallet production at MILLING. Thereby, the information network consists of three layers. Layer 1 includes the IC 'TeamCenter', i.e. the server containing specifications for the pallets. Layer 2 represents the 'transmission' layer, transferring information to the underlying systems, while Layer 3 contains 'Machine Control' for PM1-5. The production network includes five production steps: milling the basic form (PM1), drilling holes (PM2), milling T-slots (PM3), deburring (PM4), and measuring and validating the pallet (PM5). As PM2 and PM3 are in

parallel, MILLING produces two standard types of pallets, either with holes or with T-slots. As PM2 and PM3 are simple, MILLING currently produces standardised pallets with a predefined number and shape of holes and T-slots. This procedure, however, reduces production efficiency as more than the respectively required number of holes and T-slots are milled. Since PM2 and PM3 are machines of the same type, they are both controlled by 'Machine Control B'. *Scenario 2* illustrates a planned extension of the basic case in which MILLING produces customised pallets for individual customers. In this scenario, PM2 and PM3 need to be smart, as they require customer-specific information on pallet layouts. As the customised pallets contain less holes and T-slots than the standardised type, the duration of the production steps performed by PM2 and PM3 is reduced. In addition, PM5 individually measures and validates the customised pallets, making it a smart PM. *Scenario 3* differs from scenario 2 by including a backup server for the 'TeamCenter'. This is reasonable, as three out of five PMs are smart and therefore highly dependent on the availability of production-specific information.

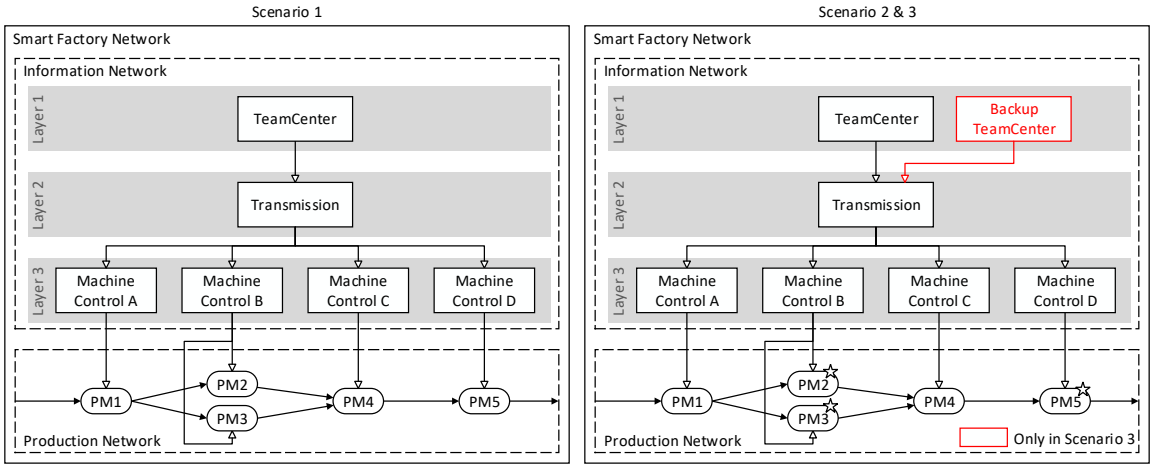


Figure 8. Smart Factory Layouts for a Real-world Use Case and Two Planned Extensions

After determining the layouts of our use cases, we simulated 1,000 cycles per scenario. Thereby, we used minutes as time unit and set the simulation cycle timeframe to 1,500 minutes, i.e. one day and one hour of transient time. To depict threat propagation within our network, the simulation is based on the following assumptions and parameters:

- For every IC, the error rate and error recovery rates are set to $\lambda_E = 0.000\overline{2}$, and $\lambda_{ER} = 0.00\overline{5}$. Thus, an error occurs every 4,500 minutes and lasts 180 minutes on average.
- For every IC, the attack recovery rate is set to $\lambda_{AR} = 0.01\overline{3}$, resulting in an average attack duration of 75 minutes per IC. ICs on different layers differ in their probability of propagation. The transmission (Layer 2) is easily accessible, while machine controls (Layer 3) are highly protected. Therefore, the propagation rates are set to $\lambda_P^{Layer2} = 0.04$, $\lambda_P^{Layer1} = 0.02$, and $\lambda_P^{Layer3} = 0.01$, resulting in mean propagation times of 25, 50, and 100 minutes, respectively.
- An adversary successfully performs an IT attack on the 'TeamCenter' server at minute 120.

Within the simulation results, we present the average of 1,000 simulation cycles. The 'availability of production machines' in Figure 9 shows that MILLING's decision to manufacture customised pallets (scenario 2 and 3) causes a significant drop in the availability of PMs in case of an attack. Further averaging the simulation results over time, the average availability of PMs in the simulated timeframe drops by 8% (scenario 2), and 5% (scenario 3), respectively. Due to the use of smart PMs, the reduced duration of the production steps, however, outweighs the unfavourable effects of the IT attack. Hence, the number of produced pallets still increases by 21% (scenario 2), and 27% (scenario 3). Comparing scenarios 2 and 3, the 'TeamCenter' backup server causes only a slight increase in availability of PMs (3%) and produced pallets (5%). This can be explained by the high propagation rate and non-redundancy of Layer 2.

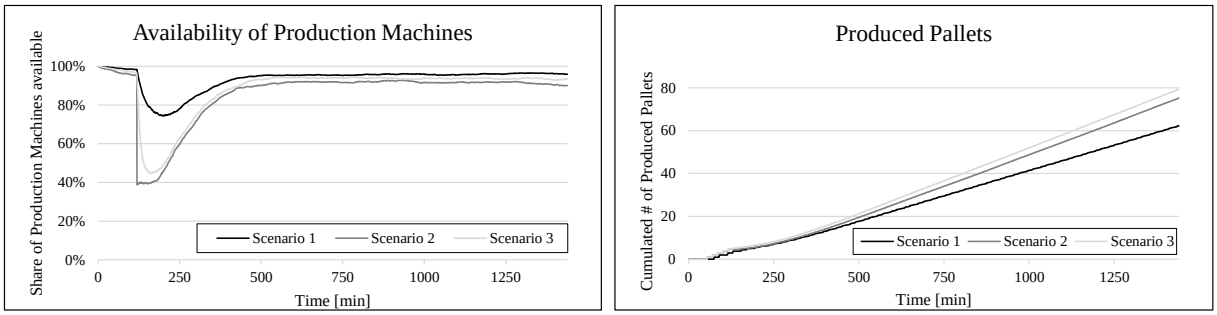


Figure 9. Performance Indicators for the Production Network

As the information networks are identical in scenario 1 and 2, the simulation shows almost the same results. Due to the 'TeamCenter' backup server in scenario 3, however, about one third of ICs remain *operational* even in case of an attack (Figure 10). Averaging these results over time, the overall functionality of ICs increases by 4% and the share of ICs in OH decreases by 48%. Focusing on the immediate effects of the attack, i.e. the first three hours after an attack, the availability of ICs increases by 22%, while the share of ICs in OH decreases by 45% due to the 'TeamCenter' backup server.

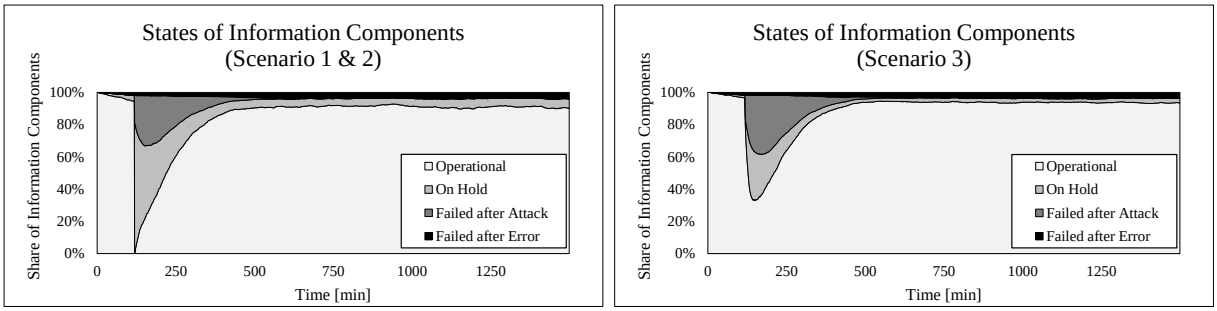


Figure 10. Performance Indicators for the Information Network

5.2 Focus Group Meeting and Expert Interview

Within our development process, we continuously evaluated our model by conducting two focus group meetings and an interview with two industry experts of MILLING. In the following, we provide selected feedback from our evaluation rounds. Being researchers in the field of SFNs and IT security, all members of our focus group confirmed the need for adequate modelling approaches for simulating and analysing impacts of threats in SFNs. Furthermore, they confirmed the use of PNs and our extensions to be adequate to depict and analyse real-world networks:

- The members criticised the initial graphical representation of the information network to be insufficient. Especially for large SFNs, the huge number of places, transitions, and arcs leads to opacity and complicates the expansion of the model. We accounted for this by including the modelling requirement of scalability (FoR.3) and applying a modular PN approach. Therefore, we developed encapsulated modules, which connect to the network via interfaces and guard functions.
- Within the second focus group meeting, all members confirmed the relevance, applicability, and usability of our model. This met the subjective ending condition of our iterative development process.

Like the focus group members, the interviewed experts confirmed the importance of IT security and, in particular, the need for modelling the propagation of attacks and errors in SFNs. MILLING already implemented several measures to improve the level of IT security as production downtimes cause great financial damage. Regarding our model, the experts understood and confirmed the suitability of our approach to model the impact of availability risks on SFNs and, thereby, support decision making regarding the layout of SFNs. Besides those general remarks, the experts provided feedback on the concrete modelling and layout:

- The experts complied with differentiating between simple and smart PMs to model the dependency on product-specific information.

- The experts pointed out that our definition of smart PMs could be extended to include machines that provide relevant information for succeeding production steps to the network, enabling concepts like predictive maintenance or real-time process optimisation. Although our PN approach is able to depict bidirectional information flows, we refrained from including them as they do not contribute to answering our research question. In addition, the mentioned concepts exceed the scope of this work. We therefore leave this extension of our model to future research.
- The experts noted that only allowing for sequential arrangement of PMs limits the applicability of our model. As the initial version of the model could not depict parallel machines without violating the requirement of scalability (FoR.3), we introduced the concept of *redundant place nodes*. By allowing for parallel machines, however, the flow of products within the production network is no longer predetermined. Therefore, we implemented a reallocation logic to transfer waiting products.
- Regarding the information network, the experts confirmed the relevance and comprehensibility of the four states of ICs. To model scheduled activities like predictive maintenance, they proposed *planned down time* as a fifth state. As we consider a rather short timeframe within our simulation, we refrained from modelling planned downtimes.

After simulating our three scenarios, we discussed our results with the experts of MILLING to confirm the usefulness of our model. Thereby, the experts agreed on the validity of our findings.

6 Conclusion, Limitations, and Future Work

Despite the increasing importance for organisations to understand and model availability risks in SFNs, the topic has been largely neglected so far. While existing works illuminate only selected aspects of the information network, production network, or failure propagation, an integrated, holistic approach is missing. Against this backdrop, we propose a modular PN approach, which enables the visual depiction and mathematical notation of scalable SFNs. In addition, it supports the modelling of specific smart factory properties, i.e. inter-modular dependencies, temporal and stochastic behaviour, product customisation, and machine parallelisation. This enables the depiction of different SFN layouts consisting of information and production components and associated dependencies. The simulation and analysis of IT attack and error propagation is enabled by using different component states, i.e. OP, OH, FE, and FA. To demonstrate the usefulness and applicability of our model, we simulated one real-world SFN and two planned expansions of a mechanical engineering company.

Our work makes two theoretical contributions: First, we provide a modular PN modelling approach which is particularly tailored towards SFN modelling by combining multiple PN approaches and elements. Second, we supplement existing research on IT security in SFNs by providing an integrated view including components, dependencies, and threat propagation from the information down to the physical machine level. Our study further entails managerial implications for practitioners. The depiction of information and production components, dependencies, and their interactions increases transparency and enables the comparison of different layouts of SFNs with regard to their susceptibility to availability risks. In addition, our approach supports risk-oriented decision making as it helps to assess risk associated with increasing interconnectivity of components, interdependencies, and redundancies in such networks. In this way, we enable the identification of critical components and information-based dependencies, which supports economically well-founded decisions on IT security mitigation measures as part of an overall IT security strategy.

As any research endeavour, our model is beset with limitations that stimulate future research. First, we limit our focus on availability risks. Future works should therefore incorporate further IT security dimensions. In contrast to availability attacks which have a direct, visible impact on a SFN, integrity and confidentiality violations initially remain unnoticed, but bear great damage potential. The need for integrating integrity and confidentiality aspects was also confirmed by our industry experts. Second, we implemented a rather simple reallocation logic to allow the distribution of waiting products. To improve production efficiency, however, advanced concepts including queuing optimisation should be integrated.

References

Amin, S., G. A. Schwartz and A. Hussain (2013). "In quest of benchmarking security risks to cyber-physical systems." *IEEE Network* 27 (1), 19–24.

Amiri, A., H. Cavusoglu and I. Benbasat (2014). "When is IT Unavailability a Strategic Risk?: A Study in the Context of Cloud Computing." In: *Proceedings of the 35th International Conference on Information Systems*. Auckland: New Zealand, p. 1–11.

Arns, M., M. Fischer, P. Kemper and C. Tepper (2002). "Supply chain modelling and its analytical evaluation." *Journal of the Operational Research Society* 53 (8), 885–894.

Billington, J. (1988). *Extending coloured petri nets*. University of Cambridge, Computer Laboratory Technical Report Number 148.

Boucher, T. O., M. A. Jafari and G. A. Meredith (1989). "Petri net control of an automated manufacturing cell." *Computers & Industrial Engineering* 17 (1-4), 459–463.

Brettel, M., N. Friederichsen, M. Keller and M. Rosenberg (2014). "How Virtualization, Decentralization and Network Building Change the Manufacturing Landscape: An Industry 4.0 Perspective." *International Journal of Information and Communication Engineering* 8 (1), 37–44.

Broy, M., M. V. Cengarle and E. Geisberger (2012). "Cyber-Physical Systems: Imminent Challenges." In: *Proceedings of the 17th Monterey Conference on Large-scale Complex IT Systems: Development, Operation and Management*. Ed. by R. Calinescu and D. Garlan. Monterey: Mexico, p. 1–28.

BSI (2017). *Cyber-Sicherheits-Umfrage 2017:* Cyber-Risiken, Meinungen und Maßnahmen. URL: https://bit.ly/2rg2EWF (visited on 11/17/2018).

Buldyrev, S. V., R. Parshani, G. Paul, H. E. Stanley and S. Havlin (2010). "Catastrophic cascade of failures in interdependent networks." *Nature* 464, 1025–1028.

Cardenas, A. A., S. Amin and S. Sastry (2008). "Secure Control: Towards Survivable Cyber-Physical Systems." In: *The 28th International Conference on Distributed Computing Systems Workshops*. Beijing: China, p. 495–500.

Cardenas, A. A., S. Amin, B. Sinopoli, A. Giani, A. Perrig and S. Sastry (2009). "Challenges for Securing Cyber Physical Systems." In: *Workshop on Future Directions in Cyber-Physical Systems Security*. Newark: New Jersey, p. 1–4.

Christensen, S. and N. D. Hansen (1993). "Coloured Petri nets extended with place capacities, test arcs and inhibitor arcs." In: *Proceedings of the 14th International Conference on Application and Theory of Petri Nets*. Ed. by M. A. Marsan. Chicago: Illinois, p. 186–205.

Danziger, M. M., L. M. Shekhtman, A. Bashan, Y. Berezin and S. Havlin (2016). "*Vulnerability of Interdependent Networks and Networks of Networks"*. 1st Edition. London: Springer.

Darwish, A. and A. E. Hassanien (2017). "Cyber physical systems design, methodology, and integration: the current status and future outlook." *Journal of Ambient Intelligence and Humanized Computing* 9 (5), 1541–1556.

De La Mota, I. F., A. Guasch, M. Mujica Mota and M. Angel Piera (2017). *Robust Modelling and Simulation*. Cham: Springer International Publishing.

Dotoli, M. and M. P. Fanti (2005). "A Generalized Stochastic Petri Net Model for Management of Distributed Manufacturing Systems." In: *44th IEEE Conference on Decision and Control & European Control*. Seville: Spain, p. 2125–2130.

D'Souza, K. A. and S. K. Khator (1994). "A survey of Petri net applications in modeling controls for automated manufacturing systems." *Computers in Industry* 24 (1), 5–16.

Dufourd, C., A. Finkel and P. Schnoebelen (1998). "Reset nets between decidability and undecidability." In: *Proceedings of the 25th International Colloquium*. Ed. by K. G. Larsen, S. Skyum and G. Winskel. Aalborg: Denmark, p. 103–115.

Fridgen, G., C. Stepanek and T. Wolf (2015). "Investigation of exogenous shocks in complex supply networks – a modular Petri Net approach." *International Journal of Production Research* 53 (5), 1387–1408.

Genge, B., I. Kiss and P. Haller (2015). "A system dynamics approach for assessing the impact of cyber attacks on critical infrastructures." *International Journal of Critical Infrastructure Protection* 10, 3–17.

Gregor, S. and D. Jones (2007). "The anatomy of a design theory." *Journal of the Association for Information Systems* 8, 312–335.

Häckel, B., F. Hänsch, M. Hertel and J. Übelhör (2018). "Assessing IT availability risks in smart factory networks." *Business Research*, 1–36.

Häckel, B., D. Miehle, S. Pfosser and J. Übelhör (2017). "Development of dynamic key figures for the identification of critical components in smart factory information networks." Working Paper.

Hermann, M., T. Pentek and B. Otto (2016). "Design Principles for Industrie 4.0 Scenarios." In: *Proceedings of the 49th Annual Hawaii International Conference on System Sciences*. Ed. by T. X. Bui and R. H. Sprague. Kauai: Hawaii, p. 3928–3937.

Interntaional Society of Automation (2010). "ANSI/ISA 95 Standard." 2010.

Ivanov, D., A. Dolgui, B. Sokolov, F. Werner and M. Ivanova (2016). "A dynamic model and an algorithm for short-term supply chain scheduling in the smart factory industry 4.0." *International Journal of Production Research* 54 (2), 386-402.

Janicki, R. and M. Koutny (1995). "Semantics of Inhibitor Nets." *Information and Computation* 123, 1–16.

Jensen, K. (1987). "Coloured Petri nets." In: *Advances in Petri nets: APN*. Ed. by G. Rozenberg. Berlin: Springer, p. 248–299.

Jensen, K. (1991). "Coloured Petri Nets: A High Level Language for System Design and Analysis." In: *High-level Petri Nets: Theory and Application*. Ed. by K. Jensen and G. Rozenberg. Berlin, Heidelberg: Springer Berlin Heidelberg, p. 44–119.

Krueger, R. A. and M. A. Casey (2014). *Focus groups:* A practical guide for applied research. 5th Edition. Thousand Oaks, California: SAGE.

Kuo, C.-H. and H.-P. Huang (2000). "Failure modeling and process monitoring for flexible manufacturing systems using colored timed Petri nets." *IEEE Transactions on Robotics and Automation* 16 (3), 301–312.

Lasi, H., P. Fettke, H.-G. Kemper, T. Feld and M. Hoffmann (2014). "Industry 4.0." *Business & Information Systems Engineering* 6 (4), 239–242.

Lee, E. A. (2008). "Cyber Physical Systems: Design Challenges." In: *Proceedings of the 11th IEEE International Symposium on Object Oriented Real-Time Distributed Computing*. Orlando: Florida, p. 363–369.

Lee, J., B. Bagheri and H.-A. Kao (2015). "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems." *Manufacturing Letters* 3, 18–23.

Long, F., P. Zeiler and B. Bertsche (2016). "Modelling the production systems in industry 4.0 and their availability with high-level Petri nets." *IFAC-PapersOnLine* 49 (12), 145–150.

Lucke, D., C. Constantinescu and E. Westkämper (2008). "Smart Factory - A Step towards the Next Generation of Manufacturing." In: *The 41st CIRP Conference on Manufacturing Systems: Manufacturing Systems and Technologies for the New Frontier*. Ed. by F. Kimura, M. Mitsuishi and K. Ueda. Tokyo: Japan, p. 115–118.

N. W. Group (2008). *Internet security glossary*. URL: http://rfc.net/rfc2828.html (visited on 11/26/2018).

Pasqualetti, F., F. Dorfler and F. Bullo (2013). "Attack Detection and Identification in Cyber-Physical Systems." *IEEE Transactions on Automatic Control* 58 (11), 2715–2729.

Peterson, J. L. (1977). "Petri Nets." *ACM Computing Surveys* 9 (3), 223–252.

Petri, C. A. (1966). "Communication with Automata." Diploma Thesis. Technical University of Darmstadt.

Radziwon, A., A. Bilberg, M. Bogers and E. S. Madsen (2014). "The Smart Factory: Exploring Adaptive and Flexible Manufacturing Solutions." *Procedia Engineering* 69, 1184–1190.

Rainer, R. K., C. A. Snyder and H. H. Carr (1991). "Risk Analysis for Information Technology." *Journal of Management Information Systems* 8 (1), 129–147.

Ramchandani, C. (1973). "Analysis of Asynchronos Concurrent Systems by Timed Petri Nets." PhD Thesis. Massachusetts Institute of Technology.

Razzaq, M. and J. Ahmad (2015). "Petri Net and Probabilistic Model Checking Based Approach for the Modelling, Simulation and Verification of Internet Worm Propagation." *PloS one* 10 (12).

Rowley, J. (2012). "Conducting research interviews." *Management Research Review* 35 (3), 260–271.

Sadeghi, A.-R., C. Wachsmann and M. Waidner (2015). "Security and privacy challenges in industrial internet of things." In: *52nd ACM/EDAC/IEEE Design Automation Conference*. San Francisco: California, p. 1–6.

Salfner, F. and K. Wolter (2009). "A Petri net model for service availability in redundant computing systems." In: *Proceedings of the 2009 Winter Simulation Conference*. Ed. by M. D. Rossetti. Austin: Texas, p. 819–826.

Schuh, G., T. Potente, R. Varandani, C. Hausberg and B. Fränken (2014). "Collaboration Moves Productivity to the Next Level." *Procedia CIRP* 17, 3–8.

Selic, B. and S. Gérard (2013). *Modeling and Analysis of Real-Time and Embedded Systems with UML and MARTE:* Developing Cyber-Physical Systems. Burlington: Elsevier Science.

Smith, G. E., K. J. Watson, W. H. Baker and J. A. Pokorski II (2007). "A critical balance: collaboration and security in the IT-enabled supply chain." *International Journal of Production Research* 45 (11), 2595–2613.

Sonnenberg, C. and J. Vom Brocke (2012). "Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research." *Design Science Research in Information Systems. Advanced in Theory and Practice. Proceedings of the 7th DESRIST Conference*. Las Vegas: Nevada, 381–397.

Szpyrka, M. and B. Jasiul (2017). "Evaluation of Cyber Security and Modelling of Risk Propagation with Petri Nets." *Symmetry* 9 (3), 1–13.

Thomesse, J.-P. (2005). "Fieldbus Technology in Industrial Automation." *Proceedings of the IEEE* 93 (6), 1073–1101.

Tremblay, M. C. and D. J. Berndt (2010). "Focus Groups for Artifact Refinement and Evaluation in Design Research." *Communications of the Association for Information Systems* 26 (1), 599–618.

Tsinarakis, G. J., K. P. Valavanis and N. C. Tsourveloudis (2003). "Modular Petri net based modeling, analysis and synthesis of dedicated production systems." In: *Proceedings of the 2003 IEEE International Conference on Robotics & Automation*. Taipei: Taywan, p. 3559–3564.

Tupa, J., J. Simota and F. Steiner (2017). "Aspects of Risk Management Implementation for Industry 4.0." *Procedia Manufacturing* 11, 1223–1230.

van Brussel, H., Y. Peng and P. Valckenaers (1993). "Modelling Flexible Manufacturing Systems Based on Petri Nets." *CIRP Annals* 42 (1), 479–484.

van der Aalst, W.M.P. (1993). "Interval timed coloured petri nets and their analysis." In: *Proceedings of the 14th International Conference on Application and Theory of Petri Nets*. Ed. by M. Ajmone Marsan. Chicago: Illinois, p. 453–472.

van der Aalst, W.M.P. (1998). "The application of Petri nets to workflow management." *Journal of Circuits, Systems and Computers* 8 (1), 21–66.

Venkatesh, K. and M. Zhou (2000). *Modeling, simulation and control of flexible manufacturing systems:* A petri net approach. Singapore: World Scientific.

Verbeek, H.M.W., M. T. Wynn, W.M.P. van der Aalst and A.H.M. ter Hofstede (2010). "Reduction rules for reset/inhibitor nets." *Journal of Computer and System Sciences* 76 (2), 125–143.

Wagner, S. M. and N. Neshat (2010). "Assessing the vulnerability of supply chains using graph theory." *International Journal of Production Economics* 126 (1), 121–129.

Wang, S., J. Wan, Di Li and C. Zhang (2016). "Implementing Smart Factory of Industrie 4.0: An Outlook." *International Journal of Distributed Sensor Networks* 12 (1), 1–10.

Wu, T., J. Blackhurst and P. O'grady (2007). "Methodology for supply chain disruption analysis." *International Journal of Production Research* 45 (7), 1665–1682.

Yoon, J.-S., S.-J. Shin and S.-H. Suh (2012). "A conceptual framework for the ubiquitous factory." *International Journal of Production Research* 50 (8), 2174–2189.

Zhang, C., S. Wang, J. Wan, D. Zhang, Di Li and C. Zhang (2016). "Towards smart factory for industry 4.0: a self-organized multi-agent system with big data based feedback and coordination." *Computer networks* 101, 158-168.

Zhu, Q., C. Rieger and T. Basar. (2011). "A hierarchical security architecture for cyber-physical systems." In: *Proceedings of the 4th International Symposium on Resilient Control Systems*. Boise: Idaho, p. 15–20.

Zuehlke, D. (2010). "SmartFactory—Towards a factory-of-things." *Annual Reviews in Control* 34 (1), 129–138.