

Modellierung und Simulation fehlerbehafteter Handlungen in Arbeitssystemen mittels Petri-Netze

Jan-Phillip HERRMANN, Sven TACKENBERG

*Labor für Industrial Engineering, TH OWL
Campusallee 12, D-32657 Lemgo*

Kurzfassung: Das Erleben der Konsequenzen von in Arbeitsprozessen verursachten Fehlern stellt eine effektive Lerngelegenheit dar (Bauer & Harteis 2012; Kapur 2015). Im Forschungsprojekt „Fehler didaktisch nutzbar machen mit Augmented Reality“ (FeDiNAR) wird ein Lernsystem entwickelt, welches fehlerhafte Handlungen von Menschen in Arbeitsprozessen erkennt und deren Konsequenzen zum Eintrittszeitpunkt in Augmented Reality (AR) simuliert. Als Formalismus für ereignisdiskrete Arbeitsprozesse lassen sich Petri-Netze für die AR-Modellierung und -Simulation nutzen (Rossmann et al. 2012). Das Erkennen der Fehlerverursachung und das Aktivieren der Visualisierung von Fehlerkonsequenzen in AR basiert auf einer handlungsbegleitenden, dynamischen Simulation der Zustände des Arbeitssystems mittels Petri-Netze. Die Abbildung relevanter Arbeitssystemelemente resultiert in einer Petri-Netz-Architektur mit einem Petri-Netz zur Modellierung der Handlung als zentrales Element.

In diesem Beitrag wird die Petri-Netz-Architektur des Lernsystems zur simulationsgestützten Beschreibung fehlerbehafteter Handlungen in Arbeitssystemen vorgestellt. Anschließend werden die Grenzen des Simulationsmodells anhand der Erfahrungen aus einem Einsatz des Lernsystems bei spezifischen Anwendungsfällen aufgezeigt und diskutiert. Der vorliegende Beitrag baut auf vorherigen Arbeiten der Autoren zum Thema Modellierung fehlerbehafteter Arbeitsprozesse mittels Petri-Netze auf (Kobelt et al. 2020b).

Schlüsselwörter: Lernsystem, Augmented Reality, ereignisdiskrete Simulation, Petri-Netz, Arbeitssystem

1. Einleitung

Im Rahmen des Forschungsprojektes „Fehler didaktisch nutzbar machen mit AR“ (FeDiNAR) wird ein Lernsystem entwickelt, welches innerhalb realer Handlungen verursachte Fehler von Arbeitspersonen beobachtet und virtuelle Fehlerkonsequenzen zum Eintrittszeitpunkt in AR visualisiert (Atanasyan et al. 2020). Das Erleben der Konsequenzen von in Arbeitsprozessen verursachten Fehlern stellt eine effektive Lerngelegenheit dar (Bauer & Harteis 2012; Kapur 2015) und wird somit ermöglicht.

Die Beobachtung der Handlungen im Lernsystem sowie die Simulation der Fehlerkonsequenzen in AR lassen sich durch Petri-Netze realisieren (Rossmann et al. 2012). Hierfür ist eine Simulation der verschiedenen Elemente des Arbeitssystems notwendig, in dem eine Arbeitsperson Handlungen ausführt. In diesem Beitrag wird ein Ausschnitt des für das Lernsystem verwendeten Petri-Netzes zur Erkennung der Fehlerverursachung und Aktivierung der Visualisierung von Fehlerkonsequenzen in

AR vorgestellt. Das nachfolgend vorgestellte Petri-Netz stellt eine Weiterentwicklung von zuvor veröffentlichten Ergebnissen dar (Kobelt et al. 2020a; Kobelt et al. 2020b). Die Architektur des Petri-Netzes stellt einen wiederverwendbaren, modularen Baustein bereit, mit dem sich der Beginn, die Ausführung sowie der Abschluss beliebiger Handlungen im Arbeitssystem modellieren und simulieren lässt.

2. Simulation von Arbeitsprozessen mittels Petri-Netze

Die Vorteilhaftigkeit der Modellierung von Arbeitsprozessen (im englischen meistens unter der Betrachtung von „workflows“) mittels Petri-Netze wurde bereits von einigen Autoren aufgezeigt (van der Aalst 2013; Adam et al. 1998; Salimifard & Wright 2001). Arbeitsprozesse lassen sich mit Hilfe von Arbeitssystemen systemtheoretisch beschreiben (Schlick et al. 2018). Als soziotechnisches System stellt ein Arbeitssystem ein komplexes System dar. Eine mit realistischem Aufwand durchführbare Simulation von Arbeitssystemen zur Erkennung und Visualisierung von Fehlern ist somit nur mit diskretem Zustandsraum und anhand von Ereignissen über die Zeit möglich (Hettinger et al. 2015). Diese Betrachtungsweise geht einher mit der allgemeineren formalen Beschreibung von Systemen wie beispielsweise in Ropohl (2009) und findet sich als Delta-Operator des zeitdiskreten Zustandswechsels im erweiterten Arbeitssystem von Schlick et al. (2018) wieder. Durch ein Petri-Netz lassen sich die Zustände und Zustandsübergänge nebenläufiger, asynchroner und verteilter ereignisdiskreter Systeme darstellen (Murata 1989). Für ein Arbeitssystem bedeutet dies, dass sich die Zustände und Zustandsübergänge von gegebenenfalls gleichzeitig ausgeführten Handlungen und der beteiligten Arbeitssystemelemente (z. B. Arbeitsmittel und -objekte) unabhängig voneinander simulieren lassen.

Die Aktivierung der Simulation von Fehlerkonsequenzen zum Eintrittszeitpunkt erfordert eine Echtzeit-Simulation des Petri-Netzes parallel zur real ausgeführten Handlung. Das Petri-Netz lässt sich demnach als echtzeitbasierter deskriptiver Ansatz zur Handlungserkennung einordnen. Hierarchische Ansätze der Handlungserkennung setzen atomare Handlungen des Menschen („Subevents“) in eine logische und zeitliche Beziehung um auf die übergeordnete Aktivität zu schließen (Aggarwal & Ryoo 2011). In Ghanem et al. (2004) wird z. B. ein interaktives System zur Erkennung von Ereignissen in Videoüberwachungsaufnahmen beschrieben und ein Petri-Netz zur Repräsentations- und Auswertungszwecken verwendet. In Nam et al. (1999) wird ein Petri-Netz zur Darstellung von Handgesten verwendet. Das Petri-Netz dient der Zusammenführung dieser feingranularen Gesten in eine übergeordnete Handlung. Petri-Netze wurden somit bereits zur Handlungserkennung verwendet. Vorangegangene Untersuchungen zur echtzeitbasierten Handlungserkennung mittels Petri-Netze in Arbeitssystemen sind den Autoren allerdings nicht bekannt.

Bezogen auf die Abbildung und Aktivierung von Fehlerkonsequenzen in Petri-Netzen wird in Zhang et al. (2007) die Aufgabenausführung alternativer Handlungsabläufe simuliert. Fehlerwahrscheinlichkeiten und stochastische Aktivitätsdauern sowie die Berücksichtigung der erforderlichen Kompetenzen in einer autonomen Fertigungszelle charakterisieren die Dynamik der Simulation.

In Kontogiannis (2003) wird ein gefärbtes Petri-Netz zur Abbildung menschlicher Fehler innerhalb der Ausführung einer Arbeitsaufgabe verwendet. Fehler werden als Attribut eines globalen Referenzarrays auf das alle Stellen und Transitionen Zugriff haben gespeichert. Eine von sechs möglichen Fehlerkonsequenzen ist beispielsweise das verspätete Abschließen einer Aufgabe durch eine verlängerte Bearbeitungszeit,

welche als Parameteränderung einer zeitbehafteten Transition umgesetzt wird.

Keine der genannten Untersuchungen betrachtet das Erkennen und Aktivieren von Fehlern und Fehlerkonsequenzen für eine den realen Arbeitsprozess begleitende Echtzeit-Simulation. Weiterhin erlaubt die Architektur der vorgestellten Petri-Netze keine Aktivierung von Fehlerkonsequenzen für die im FeDiNAR-Projekt eingesetzte Simulationssoftware VEROSIM. Die Grenzen der vorgestellten Untersuchungen bilden den Ausgangspunkt zur Entwicklung einer eigenen Petri-Netz-Architektur, welche die projektspezifischen Anforderungen erfüllt.

3. Petri-Netz-Architektur zur echtzeitbasierten Erkennung und Aktivierung von Fehlerverursachung und Fehlerkonsequenzen in Arbeitssystemen

Das nachfolgend vorgestellte Petri-Netz wird als gefärbtes Petri-Netz innerhalb der Petri-Netz-Modellierungssoftware Snoopy (Heiner et al. 2012) implementiert. Die Modellierung des vorgestellten Petri-Netzes orientiert sich an den Formalismen für gefärbte und hierarchische Petri-Netze (Jensen 1991) und bedient sich an weiteren in Snoopy verfügbaren Modellierungselementen wie beispielsweise Inhibitor-Kanten oder Reset-Kanten (Heiner et al. 2008). Dieser Beitrag fokussiert sich auf die Anwendung bestehender Formalismen und verzichtet daher auf ihre detaillierte Darstellung.

In Abbildung 1 wird die Definition einfacher und zusammengesetzter Farbmengen sowie von Variablen und Funktionen wie sie in Snoopy erfolgt gezeigt. Jede Handlung wird durch Buchstaben identifiziert (Farbe „HandlungID“). Durch Handlung „C“ wird beispielsweise das Aufbringen eines Drehmoments mittels digitalen Drehmomentschlüssel zur Fixierung einer Spritzgusswerkzeughälfte durch Pratzen an eine Spritzgussmaschinen-Traversal identifiziert. Durch die Farbmengen „Fehlertyp“ werden in den Token des Petri-Netzes Informationen über die Ausführung einer Handlung gespeichert, z. B. das zu niedrige Aufbringen des Drehmoments („ZuNiedrig“). Die Farbmengen „Drehmoment“, „Zeit“ und „Zähler“ spezifizieren den Wertebereich zur Zählung, wie viele Zeiteinheiten (Sekunden) das korrekte Drehmoment aufgebracht wurde.

Simple/Compound Colorsets			Variables	
Name	Type	Colors/Colorsets	Name	Colorset
Dot	dot	dot	hid	HandlungID
HandlungID	enum	A,B,C,D,E,F	hid2	HandlungID
Fehlertyp	enum	Fehlerfrei, Unterbrechung, Abbruch, ZuNiedrig	hid3	HandlungID
Handlung	product	HandlungID, Fehlertyp	ft1	Fehlertyp
Drehmoment	int	0-2000	ft2	Fehlertyp
Zeit	int	0-1000	ft3	Fehlertyp
Zähler	int	0-1000	nm	Drehmoment
Drehmoment_Zeit	product	Drehmoment, Zeit	t	Zeit
Fehlerkonsequenz	enum	Beschädigung	i	Zähler

Functions			
Return Type	Function Name	Parameter List	Function Body
int	vs_get_Drehmoment	Drehmoment nm	-
int	vs_get_Simulationszeit	Zeit t	-

Abbildung 1: Definition der Farbmengen, Variablen und Funktionen.

Die zusammengesetzten Farbmengen „Handlung“ und „Drehmoment_Zeit“ bilden verschiedene einfache Farbmengen innerhalb eines Tupels ab. Die Funktion „vs_get_Drehmoment“ bildet die Abfrage des aktuellen im realen Arbeitssystem mittels

digitalen Drehmomentschlüssel gemessenen und übermittelten Drehmoments durch das Petri-Netz ab. Sie wird sowohl innerhalb des Guards einer Transition als auch als Kantenausdruck verwendet (Abbildung 3). Die Funktion „vs_get_Simulationszeit“ greift auf die aktuelle Simulationszeit in der Simulationssoftware VEROSIM zu. Dieser Zugriff ist normalerweise als Programmcode im „Function Body“ implementiert. Aufgrund der Komplexität eine solche Funktion innerhalb eines Petri-Netzes abzubilden sowie fehlender Schnittstellen der Modellierungssoftware Snoopy, wird Definition und Ausführung des „Function Body“ im vorliegenden Fall an die Simulationssoftware VEROSIM ausgelagert.

In Abbildung 2 wird der zentrale Baustein des Petri-Netzes, welcher die Handlung der Arbeitsperson abbildet, dargestellt. Im Arbeitssystem mögliche Handlungen werden in der Stelle „P1“ definiert. Im Petri-Netz sind die atomaren Handlungen bzw. die Subevents des Menschen das Starten, Durchführen, Abschließen, Unterbrechen und Iterieren einer Handlung („T1“–„T5“).

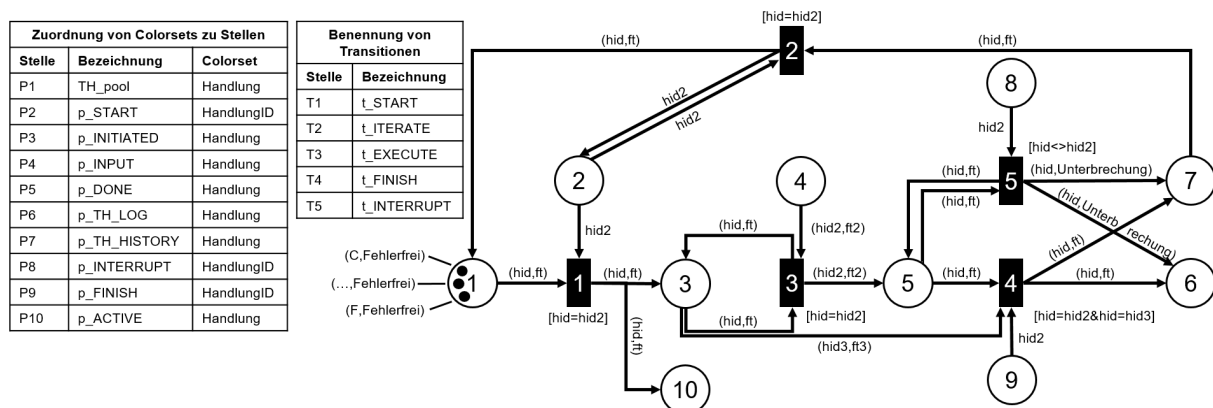


Abbildung 2: Zentraler Baustein des Petri-Netzes zur Handlungserkennung in Arbeitssystemen.

In Abbildung 3 wird das individuelle Petri-Netz des Arbeitsmittels „Digitaler Drehmomentschlüssel“ gezeigt. Die Petri-Netze der Handlung und des Drehmomentschlüssels sind über die Stellen „P2“, „P4“, „P5“, „P8“ und „P9“ verbunden. Zustandsänderungen des realen Arbeitssystems verursachen über den Zugriff der in Abbildung 1 definierten Funktionen Zustandsübergänge im Petri-Netz des digitalen Drehmomentschlüssels. Über die oben genannten Stellen werden wiederum Zustandsübergänge im Petri-Netz der Handlung verursacht. Somit wandern Token durch das Petri-Netz der Handlung hindurch, bis sie in die Stellen „P6“ und „P7“ gelangen. „P6“ dokumentiert die gesamte Historie von Handlungen als Logging. „P7“ beinhaltet ebenfalls alle durchgeführten Handlungen als Historie, allerdings lassen sich diese für eine erneute Durchführung über die Transition „T2“ entnehmen.

Das Erkennen eines Fehlers erfolgt dezentral im Petri-Netz des digitalen Drehmomentschlüssels an der Kante von „T8“ zu „P4“. Mindestens zwei Zeitschritte müssen mit einem Drehmoment von mindestens 14,00 Nm gezählt worden sein, um das Aufbringen des Drehmoments fehlerfrei ausgeführt zu haben. Andernfalls wird der Fehlertyp „ZuNiedrig“ im Token der Handlung „C“ gespeichert.

Die Aktivierung der Fehlerkonsequenz erfolgt mit dem in Abbildung 4 dargestellten Petri-Netz. Wurde die Handlung „C“ nicht fehlerfrei ausgeführt, d. h. die Spritzgusswerkzeughälfte an der Maschinentraverse mit einem zu geringen Drehmoment befestigt, könnte diese verrutschen. Beim Zusammenfahren der Werkzeughälften (Handlung „F“) könnte nun eine kritische Beschädigung der Führungsbolzen entstehen. Diese Fehlerkonsequenz wird nachdem Handlung „F“ begonnen aber noch nicht

durchgeführt oder abgeschlossen wurde vom Petri-Netz der Fehlerkonsequenz erkannt. Ein Petri-Netz für das Arbeitsmittel, an dem die Handlung „F“ durchgeführt wird (z. B. ein Bedienpanel), ist entsprechend zu erstellen. Befindet sich ein Token in der Stelle „P11“ wird eine Visualisierung der Fehlerkonsequenz durch AR in VEROSIM aktiviert.

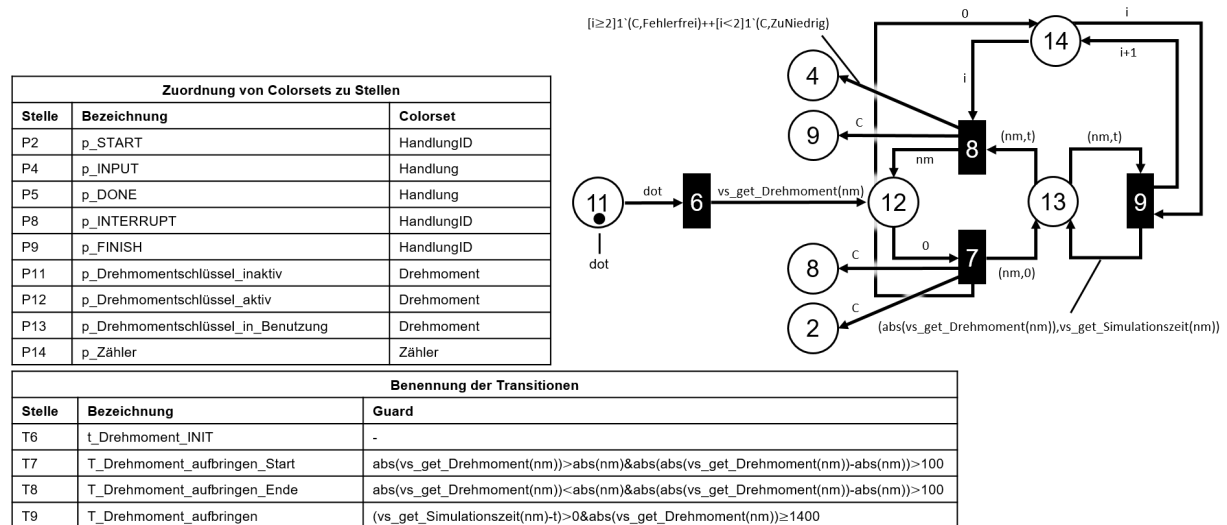


Abbildung 3: Als Petri-Netz modelliertes Arbeitsmittel „Digitaler Drehmomentschlüssel“.

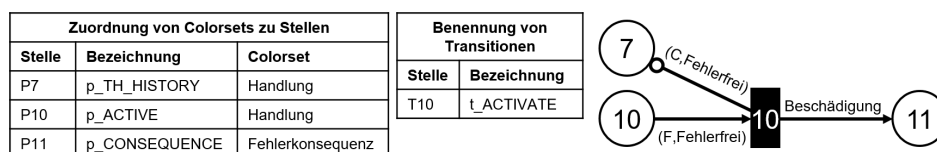


Abbildung 4: Petri-Netz der Fehlerkonsequenz „Beschädigung“.

4. Diskussion

Ein Nachteil der vorgestellten Petri-Netz-Architektur ist der hohe Modellierungsaufwand und die erforderliche Kenntnis des Modellierers. Weiterhin erfordert die hohe Komplexität des Petri-Netzes einen hohen Rechenaufwand um in der Modellierungssoftware Snoopy simuliert zu werden. Zur Abbildung realer Arbeitsprozesse in Arbeitssystemen ist der Ausgangszustand des betrachteten Arbeitssystems zu definieren und muss bei Beginn einer Simulation vorliegen oder hergestellt werden. Außerdem sind Vereinfachungen des Arbeitssystems notwendig, um eine zuverlässige Handlungserkennung zu gewährleisten. Z. B. kann das Erkennen von Beginnen und Abschließen einer Handlung mit einem Arbeitsmittel erfordern, dass das Arbeitsmittel zwingend an dessen Ausgangsposition zurückgelegt werden muss. Das Petri-Netz der Handlung ermöglicht das Erkennen von Handlungen und Fehlerverursachung in Arbeitssystemen sowie das Aktivieren virtueller Fehlerkonsequenzen noch vor Eintreten in der Realität. Nach dem Kenntnisstand der Autoren ist es das erste Petri-Netz, welches zur echtzeitbasierten Handlungserkennung in Arbeitssystemen entwickelt worden ist. Das Petri-Netz der Handlung stellt eine Menge an Stellen bereit, an die verschiedene Petri-Netze weiterer Arbeitssystemelemente angefügt werden können. Wurde ein Petri-Netz für ein Arbeitssystemelement erstellt, kann dieses wiederverwendet und frei mit anderen kombiniert werden. Die in diesem Beitrag vorgestellte Petri-Netz-Architektur wird derzeit für einen realen Anwendungsfall der beruflichen Bildung implementiert.

5. Literatur

- Adam NR, Atluri V, Huang W-K (1998) Modeling and Analysis of Workflows Using Petri Nets. In: Journal of Intelligent Information Systems 10 (2):131-158.
- Aggarwal JK, Ryoo MS (2011) Human activity analysis. In: ACM Comput. Surv. 43 (3):1-43.
- Atanasyan A, Kobelt D, Goppold M, Cichon T, Schluse M (2020) The FeDiNAR Project: Using augmented reality to turn mistakes into learning opportunities. In: Augmented Reality in Education, Springer, Cham, 71-86.
- Bauer J, Harteis C (2012) Human Fallibility. The Ambiguity of Errors for Work and Learning: Springer Science & Business Media.
- Ghanem N, DeMenthon D, Doermann D, Davis L (2004) Representation and Recognition of Events in Surveillance Video Using Petri Nets. In: 2004 Conference on Computer Vision and Pattern Recognition Workshop: IEEE.
- Heiner M, Herajy M, Liu F, Rohr C, Schwarick M (2012) Snoopy – A Unifying Petri Net Tool. In: International Conference on Application and Theory of Petri Nets and Concurrency: Springer, Berlin, Heidelberg, 398-407.
- Heiner M, Richter R, Schwarick M, Rohr C (2008) Snoopy-a tool to design and execute graph-based formalisms.
- Hettinger LJ, Kirlik A, Goh YM, Buckle P (2015) Modelling and simulation of complex sociotechnical systems: envisioning and analysing work environments. In: Ergonomics 58 (4):600-614.
- Jensen K (1991) Coloured Petri Nets: A High Level Language for System Design and Analysis. In: High-level Petri Nets: Springer, Berlin, Heidelberg, 44-119.
- Kapur M (2015) Learning from productive failure. In: Learning: Research and Practice 1 (1):51-65.
- Kobelt D, Goppold M, Atanasyan A, Herrmann J-P, Tackenberg S, Frenz M, Gamber T (2020a) Use-Case Studie eines auf der Nutzung von Handlungsfehlern basierenden AR-Lernsystems zur kritischen Reflexion der technischen Umsetzbarkeit. In: Workshop Gemeinschaften in Neuen Medien (GeNeMe) 2020. TUDpress.
- Kobelt D, Herrmann J-P, Tackenberg S, Gamber T (2020b) Petri-Netz Architekturen zur Modellierung von menschlichen Fehlern und Fehlerkonsequenzen in Arbeitsprozessen. In: Digitale Arbeit, digitaler Wandel, digitaler Mensch? 66. Kongress der Gesellschaft für Arbeitswissenschaft. GfA-Press, Berlin.
- Murata T (1989) Petri nets: Properties, analysis and applications. In: Proc. IEEE 77 (4): 541–580.
- Nam Y, Wohn N, Lee-Kwang H (1999) Modeling and recognition of hand gesture using colored Petri nets. In: IEEE Trans. Syst., Man, Cybern. A 29 (5):514-521.
- Ropohl G (2009) Allgemeine Technologie: eine Systemtheorie der Technik: KIT Scientific Publishing.
- Rossmann J, Schluse M, Waspe R (2012) Integrating object oriented Petri nets into the active graph database of a real time simulation system.
- Salimifard K, Wright M (2001) Petri net-based modelling of workflow systems: An overview. In: European Journal of Operational Research 134 (3):664-676.
- Schlick C, Bruder R, Luczak H (2018) Arbeitswissenschaft: Springer-Verlag.
- van der Aalst WMP (2013): Business Process Management: A Comprehensive Survey. In: ISRN Software Engineering 2013:1-37.

Danksagung: Der Beitrag entstammt dem Projekt „FeDiNAR – Fehler didaktisch nutzbar machen mit AR“. Es wird mit Mitteln des Bundesministeriums für Bildung und Forschung (BMBF) innerhalb des Fachprogramms „Digitale Medien in der beruflichen Bildung“ gefördert und vom DLR Projektträger unter dem FKZ 01PV18005C betreut.



Gesellschaft für
Arbeitswissenschaft e.V.

Technologie und Bildung in hybriden Arbeitswelten

68. Kongress der
Gesellschaft für Arbeitswissenschaft

Otto-von-Guericke-Universität Magdeburg

Fraunhofer-Institut für Fabrikbetrieb und
Fabrikautomatisierung IFF, Magdeburg

02. – 04. März 2022

GfA-Press

Bericht zum 68. Arbeitswissenschaftlichen Kongress vom 02. – 04. März 2022

**Otto-von Guericke-Universität Magdeburg;
Fraunhofer-Institut für Fabrikbetrieb und -automatisierung IFF, Magdeburg**

Herausgegeben von der Gesellschaft für Arbeitswissenschaft e.V.
Sankt Augustin: GfA-Press, 2022
ISBN 978-3-936804-31-7

NE: Gesellschaft für Arbeitswissenschaft: Jahresdokumentation

Als Manuskript zusammengestellt. Diese Jahresdokumentation ist nur in der Geschäftsstelle (s. u.) erhältlich.

Alle Rechte vorbehalten.

© **GfA-Press, Sankt Augustin**

Schriftleitung: Prof. Dr. Rolf Ellegast

im Auftrag der Gesellschaft für Arbeitswissenschaft e.V.

Ohne ausdrückliche Genehmigung der Gesellschaft für Arbeitswissenschaft e.V. ist es nicht gestattet:

- den Kongressband oder Teile daraus in irgendeiner Form (durch Fotokopie, Mikrofilm oder ein anderes Verfahren) zu vervielfältigen,
- den Kongressband oder Teile daraus in Print- und/oder Nonprint-Medien (Webseiten, Blog, Social Media) zu verbreiten.

Die Verantwortung für die Inhalte der Beiträge tragen alleine die jeweiligen Verfasser; die GfA haftet nicht für die weitere Verwendung der darin enthaltenen Angaben.

Geschäftsstelle der GfA

Simone John, Tel.: +49 (0)30 1300-13003

Alte Heerstraße 111, D-53757 Sankt Augustin

info@gesellschaft-fuer-arbeitswissenschaft.de · www.gesellschaft-fuer-arbeitswissenschaft.de

Screen design und Umsetzung

© 2022 fröse multimedia, Frank Fröse

office@internetkundenservice.de · www.internetkundenservice.de