

Reihe 22

Mensch-Maschine-  
Systeme

Nr. 40

Dipl.-Ing. Patrick Bertram,  
Kaiserslautern

## Entwicklung eines kontextsensitiven, modularen Assistenz- systems für manuelle Tätigkeiten



Werkzeugmaschinen  
und Steuerungen  
TU KAISERSLAUTERN



# Entwicklung eines kontextsensitiven, modularen Assistenzsystems für manuelle Tätigkeiten

Vom Fachbereich Maschinenwesen und Verfahrenstechnik  
der Technischen Universität Kaiserslautern  
zur Erlangung des akademischen Grades

**Doktor-Ingenieur (Dr.-Ing.)**

genehmigte

**Dissertation**

von

Herrn

**Dipl.-Ing. Patrick Bertram**

aus Rockenhausen

Tag der mündlichen Prüfung: 10.11.2020

Dekan: Prof. Dr.-Ing. Tilmann Beck

Promotionskommission:

Vorsitender: Prof. Dr.-Ing. Jörg Seewig

1. Berichterstatter: Prof. Dr.-Ing. Martin Ruskowski

2. Berichterstatter: Prof. Dr. Antonio Krüger

D 386





# Fortschritt-Berichte VDI

Reihe 22

Mensch-Maschine-  
Systeme

Dipl.-Ing. Patrick Bertram,  
Kaiserslautern

Nr. 40

Entwicklung eines  
kontextsensitiven,  
modularen Assistenz-  
systems für manuelle  
Tätigkeiten



Werkzeugmaschinen  
und Steuerungen  
TU KAISERSLAUTERN

Bertram, Patrick

## **Entwicklung eines kontextsensitiven, modularen Assistenzsystems für manuelle Tätigkeiten**

Fortschr.-Ber. VDI Reihe 22 Nr. 40. Düsseldorf: VDI Verlag 2020.

176 Seiten, 85 Bilder, 12 Tabellen.

ISBN 978-3-18-304022-3, ISSN 1439-958X,

€ 62,00/VDI-Mitgliederpreis € 55,80.

**Für die Dokumentation:** Industrie 4.0 – Assistenzsysteme – Manuelle Tätigkeiten – Reparatur und Nacharbeit – Mensch-Maschine Interaktion – Intentionserkennung – Kontextsensitivität – Modulare, verteilte Systeme

Die vorliegende Arbeit wendet sich an Ingenieure und Wissenschaftler im Bereich der industriellen Produktion. Sie beschäftigt sich mit der Entwicklung eines kontextsensitiven und modularen Assistenzsystems, um Werker im Bereich der Reparatur und Nacharbeit zu unterstützen. Das Assistenzsystem fokussiert flexibel bearbeitbare und wechselnde Arbeitsprozesse, bei denen weder der Arbeitsweg noch das Arbeitsziel im Voraus bekannt sind. Grundlage hierfür sind - neben des modularen und kontextsensitiven Aufbaus – die Betrachtung von deterministischen und probabilistischen Bereichen eines Arbeitsprozesses. Das System kann sich so flexible an individuelle Arbeitsweisen oder Situationen anpassen. Um die Vorteile der flexiblen Assistenz auch in bestehenden Systemen zu erreichen wurde bei der Entwicklung eine wiederverwendbare und übertragbare Systemarchitektur konzipiert.

### **Bibliographische Information der Deutschen Bibliothek**

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter <http://dnb.ddb.de> abrufbar.

### **Bibliographic information published by the Deutsche Bibliothek**

(German National Library)

The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at <http://dnb.ddb.de>.

Dissertation Technische Universität Kaiserslautern

© VDI Verlag GmbH · Düsseldorf 2020

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 1439-958X

ISBN 978-3-18-304022-3

# Danksagung

An dieser Stelle möchte ich allen Personen meinen großen Dank aussprechen, die mich bei der Erstellung meiner Dissertation und auf dem gesamten Weg dorthin unterstützt haben. Die vorliegende Arbeit entstand begleitend zu meinen Tätigkeiten bei der *SmartFactory*<sup>KL</sup> und dem Forschungsbereich *Innovative Fabrikssysteme* des DFKI.

Meinem Doktorvater Prof. Dr.-Ing. Martin Ruskowski möchte ich daher nicht nur für die Betreuung und die Begutachtung meiner Arbeit danken, sondern auch für die übertragene Verantwortung und Freiräume bei meinen Projektarbeiten. Gerade diese Möglichkeiten haben die Erstellung dieser Arbeit und die damit verbundenen praktischen Umsetzungen erst möglich gemacht. Weiterhin möchte ich Herrn Prof. Dr.-Ing. Zühlke dafür danken, dass er meinen Einstieg in die *SmartFactory*<sup>KL</sup> ermöglicht und durch die Übertragung vieler fordernder und fördernder Aufgaben maßgeblich zu meiner Entwicklung und Erarbeitung dieser Dissertation beigetragen hat.

Ebenso gebührt mein Dank Herrn Prof. Dr. Antonio Krüger für die Begutachtung meiner Dissertation, das Interesse an meiner Arbeit und an weiterführenden gemeinsamen Forschungsarbeiten in diesem Themenbereich sowie Herrn Prof. Dr.-Ing. Jörg Seewig für den Vorsitz bei der Prüfungskommission.

Weiterhin danke ich all meinen Kollegen - seien es aktuelle oder ehemalige, seien es Mitarbeiter am Lehrstuhl WSKL, an der *SmartFactory*<sup>KL</sup> oder dem Forschungsbereich IFS - für die großartige Zeit, die vielen gemeinsamen Momente sowie die stets überragende Zusammenarbeit und Unterstützung in allen Projekten. Namentlich möchte ich hier Fabian sowie die Mitarbeiter des Teams Konstruktion und Entwicklung - Federico, Rouven, Rüdiger, Rüdiger und Nils - erwähnen, die mich teilweise von Beginn an bei allen technischen Fragestellungen unterstützt und in den letzten Monaten meiner Dissertation meine persönliche Entwicklung stark geprägt haben. Besonders hervorheben möchte ich zudem Florian, Jesko und Max für all die hitzigen, vor allem aber hilfreichen Diskussionen, für die einzigartigen Momente innerhalb und außerhalb der Arbeit sowie für die ständige gegenseitige Motivation und den Antrieb bei unseren Arbeiten.

Vielen Dank wünsche ich ebenso Benjamin, Carsten, Felix, Keran, Max, Nikolas und Yuriy, die im Laufe von Projekten oder studentischen Arbeiten große Anteile an den praktischen Umsetzungen rund um den Handarbeitsplatz der *SmartFactory*<sup>KL</sup> hatten und somit erst die Rahmenbedingungen für diese Arbeit geschaffen haben.

Ebenfalls danke ich allen meinen Freunden für die moralische Unterstützung sowie den Beistand bei der schriftlichen Ausarbeitung und bei der Korrektur der Arbeit. Insbesondere danke ich Fabienne und Laura für die kontinuierliche Hilfe, durch die viele komplexe und verschachtelte Sätze erst in eine verständliche Form gebracht wurden.

In besonderem Maße danke ich meinen Eltern Andrea und Walter sowie meiner Schwester Theresa, die einen großen Einfluss auf meine Entwicklung hatten und damit meine Persönlichkeit und meinen Ehrgeiz geprägt haben. Vielen Dank für die direkte und indirekte Unterstützung, die Ihr mir habt zukommen lassen.

Patrick Bertram

# Inhaltsverzeichnis

<b>Kurzfassung</b>	<b>VIII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Problemstellung . . . . .	2
1.2 Zielstellung . . . . .	4
1.3 Vorgehensweise . . . . .	6
<b>2 Stand der Wissenschaft und Technik</b>	<b>8</b>
2.1 Assistenzsysteme für manuelle Tätigkeiten . . . . .	8
2.1.1 Begrifflichkeiten und Einordnung von Assistenzsystemen . . . . .	9
2.1.2 Bekannte Assistenzsysteme und deren Ziele . . . . .	10
2.1.3 Kontextsensitivität von Assistenzsystemen . . . . .	16
2.1.4 Assistenzsysteme im Kontext Industrie 4.0 und Reparatur . . . . .	20
2.1.5 Zusammenfassung und Abgrenzung . . . . .	23
2.2 Grundlagen zur semantischen Modellierung des Arbeitskontexts . . . . .	24
2.2.1 Kontextmodellierung: Begrifflichkeiten und bekannte Kontextmodelle . . . . .	25
2.2.2 Kontextmodelle für das Produktionsumfeld . . . . .	28
2.2.3 Modelle zur Beschreibung manueller Tätigkeiten im Produktionsumfeld . . . . .	31
2.2.4 Modelle zur Klassifikation von Werkzeugen und Materialien . . . . .	34
2.2.5 Zusammenfassung . . . . .	35
2.3 Modellierungssprachen und Verhaltensmodelle . . . . .	36
2.3.1 Ereignisgesteuerte Prozesskette (EPK) . . . . .	36
2.3.2 Business Process Modeling Notation (BPMN) . . . . .	37
2.3.3 Endliche Zustandsautomaten . . . . .	39
2.3.4 UML-Zustandsmaschine . . . . .	40
2.3.5 Petri-Netze . . . . .	42
2.3.6 Hidden Markov Modelle . . . . .	44
2.3.7 Zusammenfassung und Zwischenfazit . . . . .	46
<b>3 Methodisches Vorgehen und Anforderungsdefinition</b>	<b>48</b>
3.1 Meta-Methodik für die Bearbeitung der Zielsetzung . . . . .	48
3.2 Abgrenzung des betrachteten Problemraums . . . . .	51
3.3 Definition der globale Anforderungen . . . . .	52

3.4	Informationsakquise lokaler, funktionaler Anforderungen . . . . .	55
3.4.1	Abstrahierter Komponentenentwurf . . . . .	57
3.4.2	Abstrahierter Systementwurf . . . . .	60
3.4.3	Abstrahierte Systemspezifikation . . . . .	65
3.4.4	Zusammenfassung lokaler und funktionaler Anforderungen . . . . .	67
<b>4</b>	<b>Synthese und Modellbildung</b>	<b>69</b>
4.1	Systemspezifikation: Prozessmodell des Zielsystems . . . . .	69
4.2	Systementwurf: Entwicklung der Systemarchitektur . . . . .	72
4.3	Komponentenentwurf: Entwicklung der Verarbeitungslogik . . . . .	76
4.3.1	Bewertung bestehender Modellierungssprachen und Verhaltensmodelle	76
4.3.2	Modellierung der Arbeitsprozessfolge . . . . .	78
4.3.3	Modellierung der kontextsensitiven Erkennung des Arbeitsschritts . . .	81
4.3.4	Definition des kombinierten Arbeitsprozessmodells . . . . .	85
4.3.5	Detaillentwurf der Teilkomponenten der Verarbeitungskomponente . .	86
<b>5</b>	<b>Umsetzung und Evaluation</b>	<b>92</b>
5.1	Anwendungsfall und Assistenzsystem der <i>SmartFactory</i> <sup>KL</sup> . . . . .	93
5.2	Umsetzung und Integration der Verarbeitungskomponente . . . . .	97
5.2.1	Vorbereitung des bestehenden Assistenzsystems . . . . .	98
5.2.2	Umsetzung des Workflow-Modells . . . . .	106
5.2.3	Umsetzung der Verarbeitungskomponente . . . . .	109
5.3	Komponententest . . . . .	114
5.4	Integrationstest . . . . .	119
5.5	Systemtest . . . . .	123
5.6	Bewertung und Evaluation des Gesamtsystems . . . . .	126
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>132</b>
6.1	Implikationen für Wissenschaft und Praxis . . . . .	134
6.2	Abschließende Bewertung . . . . .	136
<b>Anhang</b>		<b>138</b>
A	Erfassungsbögen für die Informationsakquise und Informationsaufbereitung . .	138
B	Dokumente für den Systemtest . . . . .	146
B.1	Erfassungsbogen für die Testdurchläufe . . . . .	146
B.2	Arbeitsauftrag und Anwendungsszenario . . . . .	147
B.3	Nutzungsanweisung des Assistenzsystem . . . . .	149
B.4	Systembeschreibung zur Funktionsweise des Assistenzsystems und der Verarbeitungslogik . . . . .	151

<b>Literatur</b>	<b>153</b>
Monografien und Artikel . . . . .	153
Normen und Richtlinien . . . . .	164
Hochschulschriften . . . . .	164
Betreute studentische Arbeiten . . . . .	165
Webseiten und Internetreferenzen . . . . .	166

## Kurzfassung

Heutige am Markt verfügbare Assistenzsysteme zur Unterstützung manueller Tätigkeiten im Produktionsumfeld sind meist starr aufgebaut. Dies spiegelt sich häufig in unflexiblen Arbeitsplätzen wider, die Personen selbstbestimmtes Arbeiten verwehren, indem Arbeitsschritte und Abläufe vorgegeben werden und Anpassungen des Arbeitsplatzes nicht möglich sind. Selbst Systeme, die hochflexible Arbeitsaufträge bewältigen müssen, wie in der Losgröße 1 Produktion oder im Bereich der Reparatur und Nachbearbeitung von technischen Systemen, verfügen nicht über geeignete Unterstützungssysteme oder orientieren sich an den beschriebenen starren Lösungen.

Aufgrund dieser Gegebenheiten ist das Ziel dieser Arbeit ein kontextsensitives Assistenzsystem zu schaffen, das sich flexibel und automatisch an die Arbeitsweise der nutzenden Person anpasst und eine einfache Adaption des Arbeitsplatzaufbaus ermöglicht. Hierzu werden bestehende Systeme aus Industrie und Forschung untersucht und als abstraktes Modell dargestellt. Aufbauend auf den dabei gewonnen Erkenntnissen wird eine angepasste Systemarchitektur und Funktionsweise für ein modulares, kontextsensitives Assistenzsystem entwickelt.

Kernaspekt der Funktionsweise ist die Kombination eines deterministischen Anteils zur Prüfung der korrekten Bearbeitung und eines probabilistischen Anteils, der die Anpassung an den Arbeitsweg der nutzenden Person ermöglicht. In einer nachgelagerten Umsetzung am Handarbeitsplatz der *SmartFactory*<sup>KL</sup> und mithilfe eines Reparaturszenarios wird ein Machbarkeitsnachweis für das entwickelte System geliefert sowie die Funktionsweise überprüft. Die Ergebnisse dieser Evaluation zeigen, dass die Realisierung eines modularen und adaptiven Assistenzsystems erreicht wurde, wodurch eine Nutzung in veränderlichen Arbeitsumgebungen und eine Unterstützung einer eigenbestimmten Arbeitsweise für manuelle Tätigkeiten möglich wird.



# 1 Einleitung

Die Fertigung von Produkten aber auch die Produkte selbst rücken seit einigen Jahren immer stärker in das Bewusstsein der Bevölkerung. Ging es vor 10 bis 20 Jahren noch darum, wenige Produktvarianten mit einer hohen Nachfrage zu fertigen, spielt heutzutage die Individualisierung von Produkten eine zentrale Rolle [AR11; HSB11; SWM14]. Kunden wollen zum Beispiel die Ausstattung ihres Autos selbst bestimmen oder das Design ihrer Schuhe anpassen. Die Produkte sollen sich von der Masse abheben und genau auf eine individuelle Person zugeschnitten sein. In der heutigen Zeit ist jedoch nach wie vor die Massenproduktion mit Ausrichtung auf eine stabile Marktnachfrage vorherrschend. Die steigende Nachfrage nach Kundenindividualisierung sowie die Dynamisierung der Märkte stellen produzierende Unternehmen vor immer größere Herausforderungen.

Die Ausrichtung auf eine flexiblere Produktion ist der notwendige Schritt, um den wachsenden Anforderungen einer steigenden Individualisierung aber auch von kürzer werdenden Lebenszyklen (zum Beispiel bei Smartphones) entgegenzuwirken [TTV09; WLH10]. Produktionsanlagen und deren Automatisierung müssen an die dynamische Veränderung der Produktnachfrage angepasst werden. Im Rahmen der voranschreitenden Digitalisierung werden unter dem Begriff „Industrie 4.0“ Cyber-Physische Systeme (CPS) als mögliche Lösung für diese Anpassung betrachtet [KWH13]. CPS sind eingebettete Computersysteme, die über das Internet miteinander kommunizieren und eine Schnittstelle zwischen der realen und der digitalen Welt bilden [GB12]. Die Aufgaben und Funktionen, die ein CPS übernehmen kann, bietet es als Service im Internet der Dinge (IoT) an. Andere CPS oder IT-Systeme können auf die Services zugreifen und deren Funktionen nutzen. Durch die Kombination von CPSn lassen sich Cyber-Physische Produktionssysteme (CPPS) aufbauen - das heißt Produktionsanlagen, die aus vielen Services bestehen. Aus einer starren und unflexiblen Produktion wird somit die modulare Smart Factory [Zu08; Zu10].

Trotz der Verwendung von CPSn und den Potenzialen, die sie bieten, können CPS allein den steigenden Anforderungen der Produktion nicht vollständig entgegenwirken. Viele Tätigkeiten lassen sich entweder nicht wirtschaftlich oder überhaupt nicht automatisieren. Die immer schneller und stärker voranschreitende Individualisierung von Produkten verstärkt diese Herausforderung einer wirtschaftlichen Automatisierung weiter [Mü16; Sp13]. Der Bedarf von manuellen Tätigkeiten als Alternative zur Automatisierung nimmt dadurch zu. Statt jedoch eine Aufteilung in die getrennten Bereiche „Automatisierung“ und „manuelle Montage“ vorzunehmen, sieht Industrie 4.0 die Integration in eine ganzheitliche Produktion vor [Sp13].

Assistenzsysteme übernehmen diese Verbindung zwischen manuellen Arbeiten durch Menschen sowie den automatisierten Aufgaben, die Anlagen und IT durchführen. Auf der IT-Seite agiert ein Assistenzsystem wie eine automatisierte Maschine, auf der Menschenseite ist das Assistenzsystem eine individuell an den Nutzer angepasste Unterstützung. Die Herausforderung von Assistenzsystemen ist es, diese beide Seiten mit ihren unterschiedlichen Anforderungen zu berücksichtigen [Be18b]. Das heißt, die Steuerungsaufgabe der IT-Systeme darf den Menschen nicht zum rein ausführenden Arm eines Assistenzsystems machen, sondern soll ihm ein selbstbestimmtes Arbeiten ermöglichen. Umgekehrt jedoch darf auch die Zentrierung auf eine individuelle Arbeitsweise des Menschen nicht die Integration von manueller Montage in die automatisierte Produktion beeinträchtigen.

## 1.1 Problemstellung

Heute verfügbare Assistenzsysteme für die manuelle Montage unterstützen die Menschen bei der effektiven Abarbeitung ihrer Aufgaben. Erreicht wird diese Unterstützung durch die Bereitstellung von Informationen und Hilfe in der individuell richtigen Form für den Werker. Dies betrifft die Wahl der effizienten Informationsdarstellung (zum Beispiel über Bilder oder Animationen) [Fu15; Ko14b], des geeignetsten Mediums zur Informationsbereitstellung (zum Beispiel Datenbrille oder Projektionslösungen) [FKS16] und des richtigen Detailgrads der Information (zum Beispiel Einzelschritte für Laien oder Aufgabenpakete für Experten). Ziel dieser Systeme ist die kognitive Informationsassistenz, das heißt die Anpassung der Unterstützung an die kognitiven Prozesse der Person, welche das System nutzt. Die Erfassung einzelner Aktivitäten oder des Stresslevels eines Werkers ermöglicht es dem System, sich an individuelle Personen und deren momentane Bedürfnisse anzupassen.

Im Kontext von Industrie 4.0 und einer Fertigung in sogenannter Losgröße 1 stoßen kognitive Assistenzsysteme jedoch an ihre Grenzen. Das Wissen über den gesamten Montageprozess ist in dem Assistenzsystem abgespeichert. Welche Aufgaben durchgeführt werden müssen und in welcher Reihenfolge Arbeitsschritte mit welchen Hilfsmitteln zu erledigen sind, werden vom System vorgegeben. Informationen über den Kontext des Arbeitsprozesses werden zwar erhoben, von der Assistenz jedoch vornehmlich zur Steuerung der Informationsbereitstellung verwendet [Be18b]. Die Bewertung von Prozessabfolgen und die Optimierung dieser Prozesse kann von einem Assistenzsystem nicht übernommen werden. Hierbei ist der Mensch mit seinen kognitiven Fähigkeiten und seinen Erfahrungen einem technischen System überlegen. Darüber hinaus unterscheiden sich Arbeitsweisen von Mensch zu Mensch. Arbeitsschritte werden in unterschiedlichen Reihenfolgen durchgeführt und Montagetätigkeiten in einer individuellen Form ausgeübt. Aktuelle Assistenzsysteme werden diesen Anforderungen nach Flexibilität nicht in ausreichendem Maße gerecht. Abweichungen von dem vorgegeben Arbeitsprozess müssen modelliert und abgespeichert werden, damit das System diese Abweichungen erfassen und darauf reagieren

kann. Die modellierten Arbeitspläne sind zudem an individuelle Arbeitsplätze oder Assistenzsysteme angepasst und lassen sich nicht ohne weiteres auf andere Systeme übertragen. Anwender sehen sich mit der Herausforderung konfrontiert, dass Softwareentwickler oder Experten notwendig sind, um ein Assistenzsystem an veränderte Arbeitsplätze und -situationen anzupassen. Ein ganzheitlicher Ansatz zur flexiblen und adaptiven Nutzung von Arbeitsplänen existiert bisher nicht. Eine gemeinsame Integration von „manuellen Tätigkeiten“ und „Automatisierung“ in eine gemeinsame Produktion erfolgt nur zu Teilen und birgt Verbesserungspotenziale.

Die Modularisierung und Vernetzung mithilfe von CPSn sowie die semantische und damit für Maschinen und Menschen verständliche Beschreibung von Prozessen und Informationen bieten das Potenzial, die Schwächen aktueller Assistenzsysteme auszugleichen. Die Gesamtaufgabe eines Assistenzsystems kann in Einzelaufgaben oder Funktionen, wie die Erfassung des Kontextes (zum Beispiel von manuellen Tätigkeiten) und das Bereitstellen von Informationen unterteilt werden. Diese Funktionen können als Services gekapselt und durch CPS realisiert sowie bereitgestellt werden. Assistenzsysteme werden dadurch modular und deren Funktionsumfang lässt sich flexibel und individuell an den Werker beziehungsweise an den Arbeitsplatz anpassen. Die Modellierung der manuellen Montageprozesse und der Arbeitspläne ermöglicht eine maschinelle Interpretation von Arbeitsprozessen. Das Lernen von neuen oder veränderten Arbeitsabfolgen wird möglich und implizites Wissen eines Werkers kann expliziert und für andere zur Verfügung gestellt werden. Darüber hinaus kann sich das Assistenzsystem flexibel an die Arbeitsweise des Werkers anpassen, statt ihm eine feste Arbeitsfolge vorzugeben. Es wird eine verbesserte Integration von „manuellen Tätigkeiten“ und „Automatisierung“ in eine gemeinsame Produktion erreicht.

**Während es bereits Assistenzsysteme gibt, die die kognitive Unterstützung von Arbeitskräften adressieren und die Modularisierung von technischen System mittels CPSn immer weiter voranschreitet, besteht weiterhin Forschungsbedarf in der Kombination dieser beiden Paradigmen, um effiziente und an den Menschen angepasste Unterstützungssysteme zu entwickeln. Wichtige Forschungsaspekte finden sich daher in der Modellierung von flexiblen Arbeitsprozessen und in der modularen Gestaltung von kognitiven Assistenzsystemen.**

## 1.2 Zielstellung

Das Ziel der vorliegenden Arbeit ist die Entwicklung und prototypische Umsetzung eines kontextsensitiven, modularen Assistenzsystems, das sich individuell an die Arbeitsweise und den Bearbeitungsweg der nutzenden Person anpasst. Die Grundlage hierfür bilden dynamische Arbeitspläne, welche die notwendigen Arbeitsschritte zur Durchführung des Arbeitsprozesses beinhalten und diese in einer semantischen Weise für die Nutzung in Assistenzsystemen beschreiben. In diesen Arbeitsplänen können die Zusammenhänge zwischen einzelnen Arbeitsschritten des Arbeitsprozesses und die verschiedenen Bearbeitungswege zur Durchführung des Arbeitsauftrages abgebildet werden. Hierdurch kann das Assistenzsystem mögliche Bearbeitungswege identifizieren und die nutzende Person entsprechend dem gewählten Bearbeitungsweg unterstützen. Die semantische Informationsmodellierung erlaubt eine einheitliche maschinelle Interpretation der Arbeitspläne und dadurch die Übertragbarkeit dieser Pläne zur Nutzung in anderen Assistenzsystemen oder für veränderte Arbeitssituationen. Insbesondere wird durch die einheitliche Interpretation der Arbeitspläne die Realisierung eines modularen Assistenzsystems möglich, sodass dieses je nach Arbeitssituation und Bedarf mit geeigneten Sensorsystemen und Visualisierungssystemen aufgebaut werden kann.

Zur Erreichung dieser Ziele beschäftigt sich diese Arbeit einerseits mit dem Aufbau von dynamischen Arbeitsplänen und der semantischen Informationsmodellierung dieser Pläne sowie andererseits mit der Entwicklung einer Systemarchitektur zur Nutzung dieser Arbeitspläne und zur Realisierung eines modularen Aufbaus. Hierzu erfolgt eine Untersuchung der Methoden zur Prozessmodellierung, mit denen kausale und konstruktionsbedingte Reihenfolgen zwischen Arbeitsschritten abgebildet werden können und dennoch eine flexible Bearbeitungsreihenfolge nach den Präferenzen der bearbeitenden Person möglich ist (vgl. Forschungsfrage 1).

Um eine praktische Nutzbarkeit des Arbeitsmodells zu erreichen, muss das Assistenzsystem zu jedem Zeitpunkt erkennen, in welchem Arbeitsschritt sich die nutzende Person befindet. Daher werden die Möglichkeiten einer kontextsensitiven Erfassung der Arbeitssituation untersucht, um zu ermitteln wie ein Arbeitsschritt identifiziert werden kann und welche Informationen dazu notwendig sind. Hierzu zählen unter anderem notwendige Werkzeuge oder Materialien sowie die durchzuführende Tätigkeit. Die Ergebnisse der Untersuchung sind die Basis zur Erstellung des semantischen Informationsmodells als Ergänzung zum Arbeitsprozessmodell (vgl. Forschungsfrage 2).

Zur Entwicklung des kontextsensitiven, modularen Assistenzsystems wird in einem letzten Schritt die Architektur des Systems entworfen mit der einerseits eine Verarbeitung des Arbeitsmodells (Forschungsfrage 1) und andererseits eine Erkennung des bearbeiteten Arbeitsschritts (Forschungsfrage 2) möglich ist. Dabei erfolgt eine Betrachtung der Frage, wie die Architektur aufgebaut sein muss, um eine Umsetzung des Assistenzsystems unabhängig von der Arbeitssituation und der verwendeten Plattform zu ermöglichen (vgl. Forschungsfrage 3).

Zur Erreichung der Zielsetzung erfolgt eine Betrachtung und Beantwortung der folgenden wissenschaftlichen Fragestellungen:

1. Wie kann der Arbeitsprozess modelliert und verarbeitet werden, um ein flexibles Wechseln zwischen Arbeitsschritten und Bearbeitungsreihenfolgen zu ermöglichen, gleichzeitig aber die kausalen, konstruktionsbedingten Zusammenhänge des Arbeitsprozesses zu wahren?
2. Wie kann der Arbeitsprozess semantisch beschrieben werden, um eine einheitliche Interpretation für Mensch und Assistenzsystem zu gewährleisten und somit die Erkennung des sich in Bearbeitung befindenden Arbeitsschrittes zu ermöglichen?
3. Wie muss die Architektur des Systems aufgebaut sein, um die Bearbeitung des Arbeitsprozesses (1) und die Erkennung des aktuell bearbeiteten Arbeitsschrittes (2) zu ermöglichen?

Außerhalb der Betrachtung dieser Arbeit liegt die Bearbeitung der Fragen „Welchen Modularisierungsgrad muss ein Assistenzsystem besitzen?“ und „Welchen Grad an Dynamik müssen Arbeitspläne vorweisen?“. Der Aufbau der dynamischen Arbeitspläne soll technologieunabhängig und losgelöst von konkreten Arbeitsplätzen und -situationen erfolgen, um eine Übertragung in verschiedene Assistenzsysteme für manuelle Tätigkeiten zu ermöglichen. Aufgrund unterschiedlicher Bearbeitungsprozesse sowie betrieblicher Anforderungen an die Produktionsprozesse ist eine allgemeingültige Aussage hinsichtlich des Modularisierungsgrads und des Grads der Dynamik nicht möglich. Zudem herrscht eine hohe Varianz und Vielzahl an IT- und Automatisierungssystemen, sodass die Entwicklung einer spezifischen Schnittstelle zwischen Assistenzsystem und anderen Systemen des Produktionsumfeldes ebenfalls kein Betrachtungsgegenstand dieser Arbeit ist.

**Zusammengefasst ist das Ziel dieser Arbeit die Erforschung und Entwicklung eines kontextsensitiven, modularen Assistenzsystems, das die Nutzung dynamischer Arbeitspläne zur Unterstützung einer individuellen Bearbeitung von Arbeitsaufträgen ermöglicht. Dies beinhaltet insbesondere die Konzeption und die Erforschung der Realisierbarkeit mittels einer prototypischen Umsetzung für den Anwendungsfall eines Reparaturszenarios.**

Potenziale, die sich aus einem solchen System ergeben, wie das selbstständige Erlernen neuer Arbeitspläne oder Bearbeitungswege, werden nicht tiefergehend betrachtet. Dies betrifft ebenso sich daraus ergebende Möglichkeiten für die (Weiter-)Entwicklung bestehender Informationssysteme.

## 1.3 Vorgehensweise

Die Arbeit wird in die übergeordneten Bereiche *Grundlagen*, *Modellbildung* sowie *Umsetzung und Evaluation* (vgl. Abbildung 1.1) untergliedert um die Forschungsfragen zu bearbeiten und die Zielsetzung der Arbeit zu realisieren.

Im ersten Bereich der Arbeit werden die technologischen und konzeptionellen Grundlagen zur Bearbeitung der Forschungsfragen und zur Entwicklung des kontextsensitiven, modularen Assistenzsystems erarbeitet. Dazu gehören die Trends und Entwicklungstendenzen von Assistenzsystemen im Kontext des Produktionsumfeldes und Industrie 4.0 sowie bestehende Vorarbeiten zu Assistenzsystemen für die Unterstützung bei manuellen Tätigkeiten (Abschnitt 2.1). Ferner werden für eine technische Umsetzung bestehende Methoden und Konzepte der Kontextmodellierung (Abschnitt 2.2) und der Prozessmodellierung (Abschnitt 2.3) analysiert.

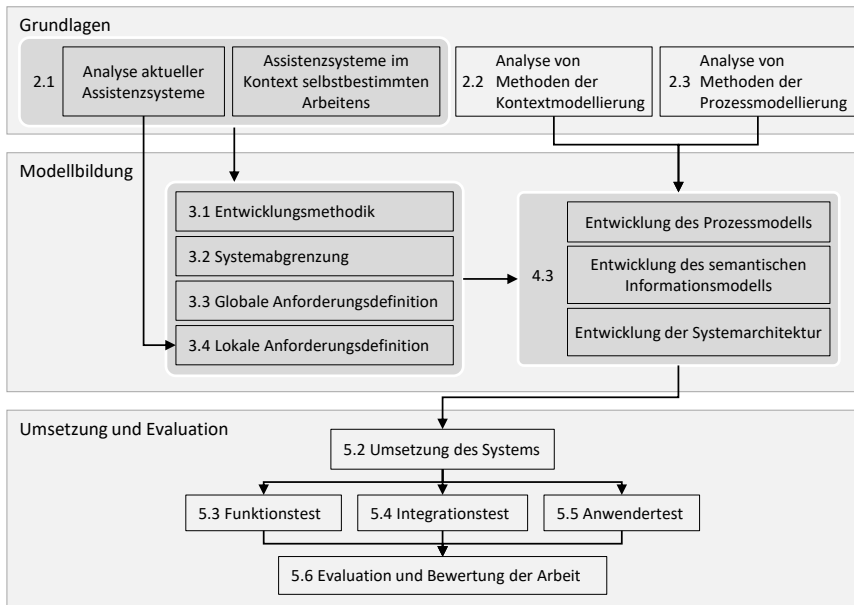


Abbildung 1.1: Vorgehensweise zur Erreichung der Zielsetzung

Im zweiten Bereich der Arbeit erfolgt die Modellierung eines kontextsensitiven, modularen Assistenzsystems auf Basis der im ersten Teil untersuchten Grundlagen. Dabei werden zuerst die methodische Vorgehensweise zur Entwicklung des Assistenzsystems erarbeitet und die Systemgrenze und die allgemeinen, globalen Anforderungen des betrachteten Systemraumes definiert (Kapitel 3). Anschließend erfolgt die Entwicklung eines abstrahierten Modells für bestehende Assistenzsysteme. Anhand dieses Modells werden funktionale Anforderungen für die Konzeptionierung und Entwicklung des Systems, als kompatible Lösung zu bestehenden Systemen,

abgeleitet (Abschnitt 3.4). Den Abschluss des zweiten Teils bilden die eigentliche Konzeption und Entwicklung des kontextsensitiven, modularen Assistenzsystems. Hierbei wird auf Basis der Grundlagen zur Prozess- und Kontextmodellierung das Arbeitsprozessmodell, das semantische Informationsmodell sowie die Systemarchitektur entwickelt (Kapitel 4).

Im dritten Bereich der Arbeit (Kapitel 5) findet die prototypische Umsetzung des entwickelten Assistenzsystems und dessen Integration in ein bestehendes System (Abschnitt 5.2) sowie eine Evaluation des entwickelten Systems (Abschnitt 5.6) statt. Die Durchführung der Evaluation basiert auf der Bewertung der Ergebnisse aus den Testszenarien

- eines automatisierten Funktionstests (Abschnitt 5.3),
- eines Integrationstests (Abschnitt 5.4) und
- eines Anwendertests (Abschnitt 5.5)

## 2 Stand der Wissenschaft und Technik

Als Hintergrundwissen für die vorliegende Arbeit werden in diesem Kapitel die technischen Grundlagen präsentiert sowie Arbeiten mit einer ähnlichen thematischen Ausrichtung vorgestellt. Hierzu wird in Abschnitt 2.1 das Aufgabenfeld der Assistenzsysteme vorgestellt. Zunächst werden wesentliche Begrifflichkeiten geklärt (2.1.1) und bekannte Assistenzsysteme aus Wissenschaft und Forschung vorgestellt (2.1.2). Anschließend wird das Konzept der Kontextsensitivität von Assistenzsystemen näher betrachtet (2.1.3) und Anforderungen für den Anwendungsbereich und das Systemumfeld erläutert (2.1.4). Das Teilkapitel endet mit einer Zusammenfassung und Abgrenzung gegenüber verwandten Arbeiten (2.1.5). Ferner werden in Abschnitt 2.2 die Grundlagen zur semantischen Modellierung des Arbeitskontexts erläutert. Hierzu gehören die Klärung relevanter Begrifflichkeiten (2.2.1), die Vorstellung bekannter Kontextmodelle im Produktionsumfeld (2.2.2) und Informationsmodelle zur Beschreibung manueller Tätigkeiten (2.2.3) sowie zur Klassifikation von Werkzeugen und Materialien (2.2.4). Nach einer Zusammenfassung der Grundlagen zur semantischen Modellierung (2.2.5) werden in Abschnitt 2.3 bekannte Modellierungssprachen und Prozessmodelle als Grundlage zur Entwicklung des Verarbeitungsmodells vorgestellt.

### 2.1 Assistenzsysteme für manuelle Tätigkeiten

Assistenzsysteme als Unterstützung des Menschen nehmen einen großen Platz in der angewandten Forschung von Mensch-Maschinen-Systemen ein. Die große Bedeutung erhalten diese Systeme einerseits durch die Wichtigkeit des Menschen als Akteur in der manuellen Montage aber auch durch die hohen Kosten, die eine manuelle Montage verursacht. So verbraucht die Montage von Maschinen und technischen Systemen nach SMITH, TAO UND HU durchschnittlich 20 % der Kosten eines Unternehmens und 50 % der Produktionszeit [Sm04; TH17]. AEHNELT UND BADER sprechen gar von 40 % der Kosten und 70 % der Zeit [AB16]. Doch trotz der hohen Kosten und der Potentiale automatisierter Systeme bleibt die manuelle Montage ein wichtiger Bestandteil der Fertigung von Produkten [Sp13]. Aufgabe und Ziel von Assistenzsystemen in der manuellen Montage ist es, Mitarbeiter bei den Entscheidungen, die sie treffen müssen, situativ zu unterstützen [Sp13]. Durch diese Unterstützung kann die nutzende Person die Aufgabe schneller und besser erledigen, wodurch die Produktionszeit verringert wird. KOKKALIS ET AL. zeigen dies beispielsweise durch eine automatische Bereitstellung von Aktionsplänen [Ko13].



### 2.1.1 Begrifflichkeiten und Einordnung von Assistenzsystemen

Unter dem Begriff Assistenzsysteme, die den Menschen unterstützen, werden viele unterschiedliche Produkte oder auch Unterstützungsmöglichkeiten zusammengefasst. Um eine Einordnung der Systeme zu erreichen haben APT ET AL. im Rahmen des Forschungsberichts 502<sup>1</sup> eine Klassifizierung von digitalen Assistenzsystemen durchgeführt [Ap18]. Grundlage ist die Einteilung entsprechend der Charakteristika zum Grad der Unterstützung, der Art der Unterstützung und der Zielsetzung der Unterstützung (vgl. Abbildung 2.1).



Abbildung 2.1: Klassifizierung von Assistenzsystemen nach [Ap18]

Bei der Art der Unterstützung entspricht dies einer Unterteilung in physische, sensorische und kognitive Systeme. Während physische Assistenzsysteme die Belastungen von körperlichen Tätigkeiten reduzieren oder gar komplett verhindern, unterstützen sensorische System bei der Sinneswahrnehmung oder Erfassung von Informationen. Kognitive Unterstützungssysteme hingegen sind charakterisiert durch eine situations- und nutzerangepasste Bereitstellung von Informationen und Hilfeleistungen. Auch wenn sich durch diese Einteilung bereits eine grobe Zuordnung zu aufgabenspezifischen Zielen ergibt, können Assistenzsysteme ferner hinsichtlich ihrer Zielsetzung eingeteilt werden. APT ET AL. sprechen hierbei von fähigkeitskompensierenden, -erhaltenden oder -erweiternden Systemen. Das Ziel von kompensatorischen Systemen ist der Ausgleich von fehlenden oder verringerten Fähigkeiten jeglicher Art auf Grund von körperlichen oder geistigen Beeinträchtigungen. Fähigkeitserweiternde Systeme hingegen zielen auf das Trainieren und Ausbilden von Fähigkeiten ab. Hinsichtlich des Grades der Unterstützung unterscheiden sich die Systeme darin, ob sie nur bei einfachen und repetitiven Situationen unterstützen (Niedrig) oder mit Formen von Künstlicher Intelligenz (KI) ausgestattet sind und so komplexe Situationen bewerten sowie Handlungsempfehlungen geben können (Hoch). Modulare und intelligente Assistenzsystemen, die sich an die Bedürfnisse der nutzenden Person und deren Unterstützungsbedarf anpassen, werden in die Gruppe der System mit variablem Grad der Unterstützung eingeordnet.

<sup>1</sup>Studie im Auftrag des Bundesministeriums für Arbeit und Soziales zum Einsatz von digitalen Assistenzsystemen

### 2.1.2 Bekannte Assistenzsysteme und deren Ziele

Im Nachfolgenden werden die bekanntesten Assistenzsysteme aus der Wissenschaft und Forschung vorgestellt und ihre jeweiligen Besonderheiten und Forschungsfokusse näher erläutert. Darüberhinausgehend existieren weitere industrielle Systeme zu denen in erster Linie Produktbeschreibungen und Marketinginformationen öffentlich verfügbar sind. Eine nähere Darstellung dieser Systeme erfolgt an dieser Stelle daher nicht. Jedoch werden wichtige Informationen über diese sowie die im nachfolgenden dargestellten Systeme mithilfe von Erfassungsbögen gesammelt und in einheitlicher Form in Anhang A repräsentiert. Die Bögen erheben dabei keinen Anspruch auf eine vollständig genaue Abbildung aller Aspekte der untersuchten Systeme. Dies liegt einerseits daran, dass die Bögen primär die Kernaspekte und Unterschiede der Systeme identifizieren sollen und andererseits daran, dass aufgrund der oben genannten Gründe für viele der Systeme kein tiefergehender technischer Einblick besteht. Eine vollständige Repräsentation aller untersuchten Systeme ist daher nicht möglich. Die Erfassungsbögen beschränken sich stattdessen auf eine abstrakte Ebene, die die Kernaspekte der Systeme und deren Besonderheiten widerspiegeln. Abbildung 2.2 zeigt beispielhaft einen ausgefüllten Erfassungsbogen für das ACTIVEASSIST-System der Firma Bosch Rexroth AG. Eine weiterführende Beschreibung zum Aufbau und zur Verwendung der Erfassungsbögen ist in Abschnitt 3.4 (Informationsakquise lokaler, funktionaler Anforderungen) zu finden.

#### System

## ActiveAssist

Bosch Rexroth AG

#### Quellen

Literaturrecherche  
Messebesuche  
Forschungsprojekt DAKARA

#### Erfassung (Eingabe)

- Handerkennung
  - Ultraschall (Stabschrauber)
  - RFID-/Barcode-Leser
- Erweiterbar

#### Visualisierung (Ausgabe)

- Monitore
  - Projektor
  - Pick-to-Light
- Erweiterbar



Bildquelle: Bosch Rexroth AG

#### Prozessverarbeitung

Vergleich von eingehenden Sensorwerten mit erwarteten Werten aus Prozessmodell

#### Prozessmodell

- Deterministisch verkettete Arbeitsschritte
- Ver. Erfahrungslevel (Aufgabe/Unteraufgaben)

#### Besonderheiten

- Modulares webbasiertes System
- Erweiterbar
- Qualitätskontrolle

Abbildung 2.2: Ausgefüllter Erfassungsbogen für Assistenzsysteme der manuellen Montage, am Beispiel ActiveAssist (Bildquelle: Bosch Rexroth AG)



formationen der Arbeitssituation, wie zum Beispiel die Entnahme von Werkzeugen oder des derzeitigen Aufenthaltsorts, und der Bewertung der Historie aller erfassten Informationen bestimmt das System die wahrscheinlichste Arbeitssituation. Um die Nutzbarkeit solcher Assistenzsysteme nicht nur unter Laborbedingungen, sondern auch unter industriellen Bedingungen zu gewährleisten, ist die Art und Weise wie relevante Informationen der Arbeitssituation erfasst werden von zusätzlicher Bedeutung. AEHNELT ET AL. beschäftigen sich daher mit der Nutzung von nicht invasiven Sensorsystemen und wie eine adäquate Erfassung ohne am Körper getragener Sensoren möglich ist [AGU14].

### MotionEAP

Ein weiteres Beispiel kognitiver Assistenzsysteme findet sich in der Unterstützung von Personen mit beispielsweise körperlichen oder geistigen Einschränkungen. BÄCHLER, FUNK, KORN U. A. beschäftigen sich mit Assistenzsystemen, zum Beispiel für ältere Mitarbeiter, zum Teil mit körperlichen Einschränkungen, die sich auf dem heutigen Arbeitsmarkt aufgrund des demografischen Wandels vermehrt wiederfinden [Bä15; Fu15; KSH13]. Im Rahmen des MotionEAP-Projektes und darauf aufbauender Arbeiten wird sich mit den Fragen beschäftigt, wie verschiedene Ansätze der Visualisierung von Informationen oder die ergonomische Anpassung des Arbeitsplatzes zu einer verbesserten Arbeitssituation für Menschen mit Beeinträchtigung führen können [motionEAP 1]. In diesem Kontext vergleichen KORN ET AL. die Effekte einer situationsabhängigen Projektion von Informationen auf den Arbeitsbereich oder der Nutzung von Gamification-Elementen bei der bildschirmgestützten Anzeige von Informationen [Ko14b]. Insbesondere werden die Einflüsse auf die Arbeitsgeschwindigkeit, -qualität und den Lernerfolg untersucht. Als Ergebnis seiner Arbeit stellt er ein Konzept für die Nutzung von Projektion und

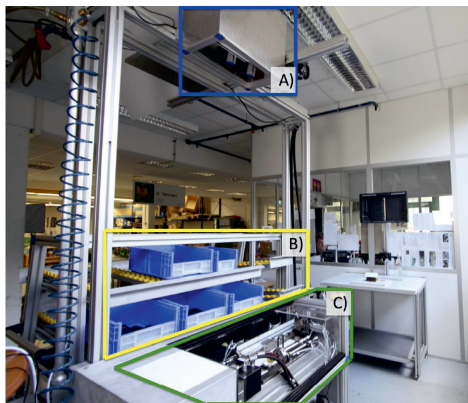


Abbildung 2.4: Handarbeitsplatz für die manuelle Montage mit Tiefenbildkamera und Projektor (A), Materialbereitstellung (B) und Arbeitsbereich (C) (Quelle: [Fu15])

Gamification in kontextsensitiven Assistenzsystemen (CAAS) vor [Ko14a]. Darüber hinaus beschäftigt sich FUNK mit verschiedenen visuellen Darstellungen von Arbeitsanweisungen und wie stark diese die nutzenden Personen unterstützen [Fu15; Fu16a]. Die Forscher untersuchen dabei Arbeitsanweisungen in Form von Bildern, Videos oder Konturen, die relevante Gegenstände oder deren Position hervorheben. Abbildung 2.4 zeigt den Aufbau des verwendeten Systems, wobei sich der Projektor über der Materialbereitstellung und dem Arbeitsbereich befindet.

### **cubu:S**

Aufbauend auf den Erfahrungen aus dem motionEAP-Projekt und dem Demonstrationsaufbau entwickelte die Firma Schnaithmann Maschinenbau GmbH anschließend das Werkerassistenzsystem cubu:S, das als marktreife Lösung angeboten wird (vgl. Abbildung 2.5) [@Schnaithmann]. Verwendung findet das System als Erweiterung für Handarbeitsplätze in der manuellen Montage. Die Prinzipien der In Situ Projektion und der visuellen Unterstützung der Montage über eine Schritt-für-Schritt Anleitung wurden in cubu:S übernommen, wobei zusätzliche Funktionen wie eine Roboteranbindung für kooperatives Arbeiten integriert wurden.



Abbildung 2.5: Werkerassistenzsystem cubu:S für Arbeitsplätze der manuellen Montage (Quelle: [@Schnaithmann])

### **MonSiKo**

Im Rahmen des Projektes MonSiKo („Adaptives Montageassistenz- und Interaktionssystem mittels 3D-Szenenanalyse und intuitiver Mensch-Technik-Kommunikation“) wurde ein Assistenzsystem entwickelt, das den Fokus weniger auf die visuelle Bereitstellung von Informationen legt, sondern auf die Erfassung der aktuellen Arbeitssituation und Tätigkeiten, die die arbeitende Person durchführt [@MonSiKo]. Kernelemente des Systems sind eine Tiefenbildkamera zur 3D-Szenenerfassung und akustische Sensoren zur Erfassung von charakteristischen Geräuschen

des Arbeitsprozesses. Die Szenenanalyse erfolgt mit Methoden des maschinellen Lernens, um Handpositionen und -gesten zu detektieren, die für die einzelnen Arbeitsschritte typisch sind [RJ19]. Durch die Kombination beider Eingangssignale wird eine intuitive Mensch-Maschinen-Interaktion umgesetzt, die der nutzenden Person keinen starren Arbeitsplan vorgibt, sondern auf diese reagiert. Dadurch können Ablageorte für Materialien und Werkzeuge sowie die Reihenfolge der Montageschritte frei gewählt werden. Das System ist dennoch in der Lage, die Tätigkeiten zu interpretieren und beim Montagevorgang zu unterstützen.



Abbildung 2.6: MonSiKo-Demonstrator als Exponat auf der Fachmesse „Control“ 2018 (Quelle: [MonSiKo])

Darüber hinaus ist ein kollaborativer Roboterarm in das MonSiKo-System integriert, der die kooperative Bearbeitung des Montageauftrages ermöglicht. Repetitive und leicht zu automatisierende Aufgaben können durch die Mensch-Maschinen-Schnittstelle an den Roboterarm delegiert und abgegeben werden. Abbildung 2.6 zeigt den Demonstrator des MonSiKo-Projektes als Exponat auf der internationalen Fachmesse für Qualitätssicherung „Control“ im Jahre 2018. Auf der linken Seite befindet sich ein Touchdisplay zur Visualisierung von Informationen und der direkten Interaktion mit dem System. Der kollaborative Roboter befindet sich hinter dem Arbeitsbereich der nutzenden Person. Im rechten Bereich ist eine Ultraschallschweißstation verbaut, mit der der Roboterarm interagiert.

### Handarbeitsplatz der *SmartFactory*<sup>KL</sup>

Ein wiederum anderer Fokus wurde beim Assistenzsystem am Handarbeitsplatz der *SmartFactory*<sup>KL</sup> verfolgt. Das System integriert einen Arbeitsplatz für manuelle Montageprozesse in den Fertigungsprozess eines modularen Fabriksystems. Kernelement der Gesamtanlage ist die Fertigung auf Basis eines Produktgedächtnisses, das alle Aufgaben und Arbeitsschritte von

Bearbeitungsbeginn bis zum fertigen Produkt beinhaltet [We15; We17b]<sup>2</sup>. Jede Fertigungsstation, die ein Teilprodukt empfängt, gleicht den aktuellen Arbeitsschritt entsprechend dem Produktgedächtnis mit den eigenen Fähigkeiten ab und führt je nach Möglichkeiten den Arbeitsschritt durch oder schickt das Produkt weiter. Existieren Arbeitsschritte, für die es keine automatisierte Fertigung gibt - zum Beispiel aufgrund von Wartungsaufgaben an der Anlage oder weil die Tätigkeiten nicht wirtschaftlich zu automatisieren sind - erfolgt eine Bearbeitung am Handarbeitsplatz. Durch die Informationen zum Fertigungsprozess im Produktgedächtnis weiß das Assistenzsystem um welches Produkt es sich handelt und welcher Arbeitsschritt durchgeführt werden muss.



Abbildung 2.7: Handarbeitsplatz mit Assistenzsystem der SmartFactory<sup>KL</sup>

Der in Abbildung 2.7 dargestellte Handarbeitsplatz bildet das zu Grunde liegende Basissystem für diese Arbeit, welches für die Entwicklung und die prototypische Umsetzung verwendet wurde. Der Arbeitsplatz ist mit Monitoren und einem Projektor sowie mit einer Tiefenbildkamera und einem RFID-Reader<sup>3</sup> ausgestattet. Über den Projektor und die Monitore erfolgt die Unterstützung der arbeitenden Person mittels einer Informationsbereitstellung zu den Tätigkeiten des aktuellen Arbeitsschritts [Qu16]. Das System zeigt in einer Schritt-für-Schritt Anleitung, wo sich Werkzeuge und Materialien befinden und wie der Montagevorgang durchzuführen ist. Die Erfassung der Arbeitssituation erfolgt über die Bestimmung der Handposition mit der Tiefenbildkamera. Das Produktgedächtnis wird durch das System mittels RFID-Reader aus dem Produkt ausgelesen.

<sup>2</sup>vgl. Herstellerunabhängige und modulare Industrie 4.0 Anlage in Abschnitt 2.1.4

<sup>3</sup>vgl. Radio Frequency Identification (RFID) in [TT10]. RFID ist eine Technologie zur kontaktlosen Kommunikation zwischen einem Sender und Empfänger, um Daten und Informationen auszutauschen.

### 2.1.3 Kontextsensitivität von Assistenzsystemen

Die Gemeinsamkeit all dieser dargestellten Systeme ist die Erfassung oder Wahrnehmung von verschiedenen Aspekten der Arbeitssituation, wie des Fertigungsprozesses, des mentalen Zustandes der nutzenden Person, des Arbeitsortes oder der Verwendung von Werkzeugen und Arbeitsmaterialien. Je mehr Aspekte das System erfasst, desto genauer kann es die tatsächliche Arbeitssituation bestimmen oder Abschätzungen zu dieser durchführen. Auf Basis der erfassten Arbeitssituation trifft das System Entscheidungen zur Bereitstellung und der Form von Unterstützungsfunktionen, zum Beispiel der Informationsbereitstellung zu einer Montagetätigkeit oder der kollaborativen Zusammenarbeit mittels eines Roboterarms. Da die Systeme auf ihren eigenen Kontext und den Kontext zur Arbeitssituation reagieren, werden diese als kontextsensitive Assistenzsysteme oder *context aware assitive systems* (CAAS) bezeichnet [Ko14b].

Insbesondere für Systeme, die mit dem Menschen interagieren und diesem helfen sollen, existiert ein hoher Bedarf an einer angepassten Kontextsensitivität, um tatsächlich eine Unterstützung zu bieten und nicht gar das Gegenteil zu bewirken. Eine kompromisslose Führung durch den Arbeitsauftrag ohne Berücksichtigung der Arbeitssituation und der nutzenden Person, führt zwangsläufig zu suboptimalen Arbeitsprozessen durch die arbeitenden Person [Wi14]. Ein gutes Beispiel für diesen Effekt sind Systeme, die auf eine feste Nutzergruppe (zum Beispiel Rechtshänder) ausgerichtet sind und die Unterschiede, die sich für andere Nutzergruppen (Linkshänder) ergeben, nicht berücksichtigen. Benutzt ein Linkshänder das System, muss dieser die Prozesse gegebenenfalls in einer anderen Art und Weise durchführen und der Effekt der Unterstützung geht verloren. Ein weiteres Beispiel ist die Unterstützung von Laien und Experten mit einem System, das weder eine Unterscheidung der Bedürfnisse der beiden Gruppen durchführt, noch erkennt, welcher von beiden Typen das System gerade nutzt. Hierdurch gehen die positiven Effekte von Assistenzsystemen verloren, da die Systeme als ungeeignet und störend empfunden werden. ZÄH UND WIESBECK sprechen bei diesem Phänomen vom Effekt des *Attention tunneling* [Zä07; ZW08]. Als Grund für die Entstehung von *Attention tunneling* gilt die Überreizung der menschlichen Wahrnehmung mit Informationen, insbesondere wenn diese keinen Bezug zur aktuellen Situation oder dem Unterstützungsbedarf der nutzenden Person haben. Beim ersten Beispielsystem für Rechtshänder kann der Effekt dazu führen, dass die Person auf keine Information oder Hilfestellung des Systems mehr reagiert, da dieses in den Augen der nutzenden Person keine Unterstützung zu liefern scheinen. Im zweiten Beispiel führt das unpassende System zu einer Überreizung der Wahrnehmung des Laien, da zu viele Informationen zum Arbeitsprozess angezeigt werden, sodass dieser überfordert ist und nicht weiß, welche Information er in welcher Weise verarbeiten soll.

Während in Abschnitt 2.1.2 bereits Technologien und Systeme vorgestellt wurden, die den vorliegenden Arbeitskontext und die Situation der nutzenden Personen erfassen, repräsentieren diese lediglich Möglichkeiten zur Akquise von Informationen. Zur Verarbeitung der Informationen, um daraus Wissen zu generieren und die vorherrschende Arbeitssituation interpretieren zu



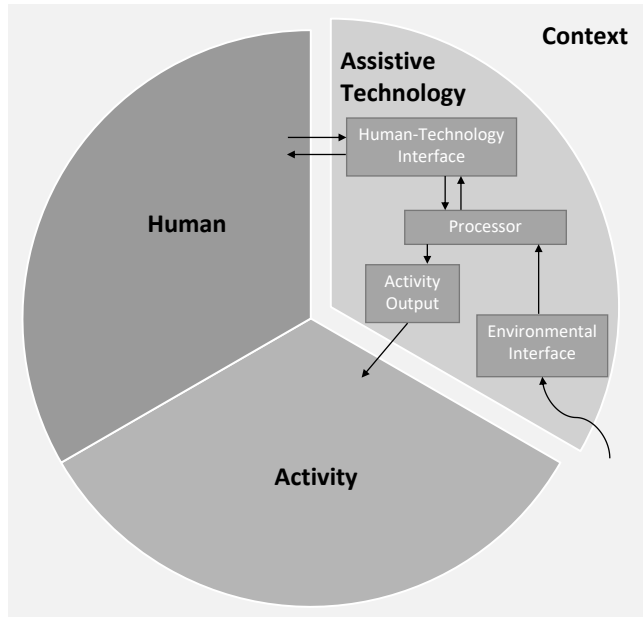


Abbildung 2.8: HAAT-Modell nach [CH02]

können, werden über die Informationsakquise hinausgehend spezielle Informations- und Verarbeitungsmodelle verwendet. Ein bekanntes Modell im Rahmen von assistierenden Systemen ist das „Human Activity Assistive Technology“-Modell (HAAT-Modell) [CH02], dargestellt in Abbildung 2.8. Das HAAT-Modell unterscheidet die vier Bereiche *Mensch*, *Aktivität*, *Kontext* und *Unterstützungstechnologien*, die miteinander in Verbindung stehen. In diesem Modell wird der Mensch mit seinen Fähigkeiten und seinem Knowhow als intrinsischer Enabler gesehen. Der Bereich der Unterstützungstechnologien, der über eine Mensch-Maschinen-Schnittstelle mit dem Menschen interagiert und die aktuelle Situation mittels eines Prozessors bewertet, wird hingegen als extrinsischer Enabler angesehen. Die Arbeitssituation und welche Arbeiten und Tätigkeiten durchgeführt werden sind durch den Bereich des Kontexts repräsentiert. Die Erfassung des aktuellen Kontexts erfolgt durch Sensoren, die als Schnittstellen zur realen Umwelt agieren (vgl. Abschnitt 2.1.3).

Ein weiteres Modell ist das „Context Aware Assistive System“-Modell (CAAS-Modell) nach KORN, das in Abbildung 2.9 dargestellt ist [Ko14a]. Das CAAS-Modell kombiniert das HAAT-Modell mit Aspekten aus dem Bereich der Spieleentwicklung, um eine positive Stimmung bei der nutzen-den Person und damit den mentalen Zustand des „Flows“ hervorzurufen. CSIKSZENTMIHALY bezeichnet „Flow“ als den mentalen Zustand, indem eine Person vollständig in ihre Tätigkeit vertieft ist und das Gefühl verspürt, voller Energie zu sein und die Aufgabe erfolgreich zu erledigen [Cs75].

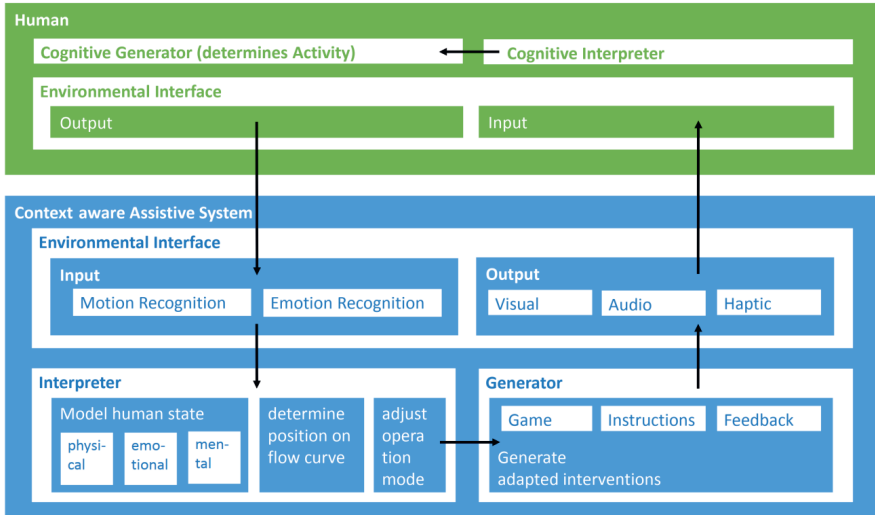


Abbildung 2.9: CAAS-Modell als Erweiterung des HAAT-Modells um Aspekte der Gamification (Quelle: [KFS15])

Das Modell zeigt insbesondere die Parallelen zwischen der Wahrnehmung der Informationen, der Interpretation der Informationen (Interpreter), der Verarbeitung (Generator) und der anschließenden Ausgabe beziehungsweise der Reaktion bei einem Menschen sowie einem CAAS. Im Vergleich zum HAAT-Modell, das die Zusammenhänge von Unterstützungssystemen sehr allgemein beschreibt, ist das CAAS-Modell spezifischer aufgebaut und auf ein konkretes Ziel ausgerichtet [KFS15].

Durch die große Bandbreite an Zielen und Aufgaben von CAAS sind mehrere Disziplinen und Themenbereiche an der Umsetzung und Weiterentwicklung dieser Systeme beteiligt. Zu den wichtigsten Disziplinen gehören der Maschinenbau und die Informatik auf der technischen Seite sowie mit einer stärker geisteswissenschaftlichen Ausrichtung die Psychologie und Ethik. Abbildung 2.10 gibt einen Überblick und eine Einordnung der Disziplinen sowie deren Technologien und Teilbereiche. Im Rahmen dieser Arbeit werden primär die Bereiche Informationsmodellierung, Prozesssteuerung und die Bereiche der Cyber-Physischen Systeme (CPS) und Cyber-Physischen Produktionsmodule (CPPM) betrachtet und im weiteren Verlauf erläutert, um ein besseres Verständnis zu schaffen und die Grundlagen für diese Arbeit bereitzustellen.

Ziel von Informationsmodellen ist die einheitliche Repräsentation von Informationen und deren Zusammenhänge, um diese interpretieren und auswerten zu können. Der Prozess der Informationsmodellierung beschäftigt sich daher damit, wie das Wissen eines spezifischen Bereiches der realen Umwelt abgebildet und in ein repräsentatives Modell überführt wird. Im Bezug zu CAAS für manuelle Montagetätigkeiten entspricht dies der Modellierung der Arbeitssituation

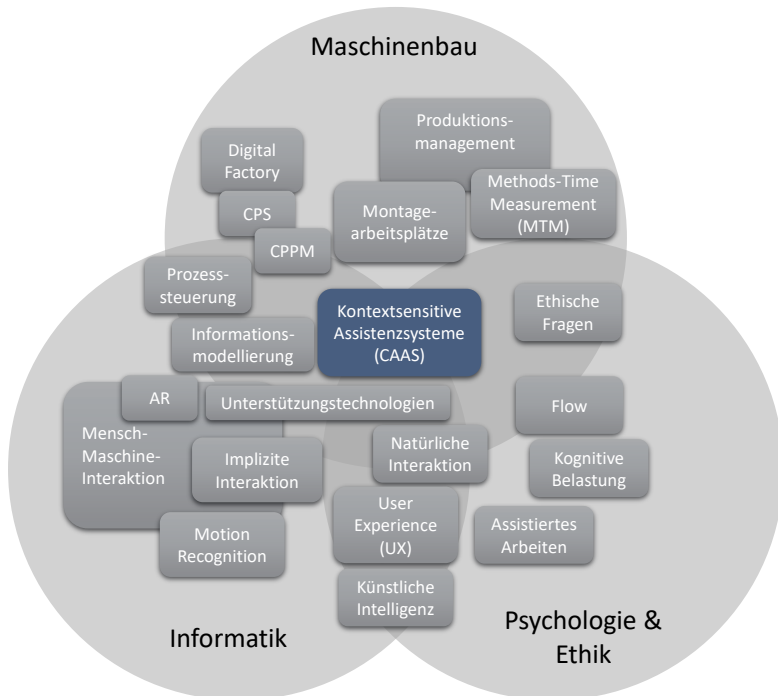


Abbildung 2.10: Übersicht der an CAAS beteiligten Disziplinen und Technologien in Anlehnung an [Ko14a]

und der durchzuführenden Arbeitstätigkeiten, für die das System Unterstützungsleistungen erbringt. In Abschnitt 2.2 werden bekannte Kontextmodelle der manuellen Montage beleuchtet und für diese Arbeit relevante Einflüsse auf die Modellierung des Arbeitskontexts erläutert.

Abschnitt 2.3 beschäftigt sich mit den Möglichkeiten zur Modellierung und Steuerung von Arbeitsprozessen und Arbeitsabläufen. Als Prozess wird im Rahmen dieser Arbeit die Verkettung und der Übergang zwischen Arbeitsschritten des Arbeitsprozesses bezeichnet. Hintergrund der Modellierung des Arbeitsprozesses und Steuerung durch diesen Prozess ist die Bestimmung der aktuellen Arbeitssituation, die sich

1. aus dem aktuellen Zustand (vgl. Kontextmodellierung in Abschnitt 2.2) und
2. aus der Abfolge von Arbeitsschritten ergibt, die zu diesem Zustand geführt haben.

Die Betrachtung von CPS und CPM erfolgt in Abschnitt 2.1.4, im Rahmen der Entwicklungstendenzen von Industrie 4.0 und der Produktionsautomatisierung. Durch diese Entwicklungen verändert sich die Art und Weise wie automatisierte Maschinen aber auch Mensch-Maschinen-Systeme arbeiten und miteinander kooperieren. Im Rahmen dieser Arbeit werden diese Ent-

wicklungstendenzen und die Anforderungen, die sich daraus an CAAS ergeben, analysiert und bei der Umsetzung einer kontextsensitiven Verarbeitungslogik berücksichtigt.

### 2.1.4 Assistenzsysteme im Kontext Industrie 4.0 und Reparatur

Eine große Herausforderung, der sich heutige Unternehmen gegenübersehen, ist die steigende Nachfrage nach Produkten mit immer stärkerer Kundenindividualisierung. Dieser Trend führt dazu, dass eine Fertigung von Produkten nach den Prinzipien der Massenproduktion oder der variantenreichen Serienproduktion immer weniger im Fokus der Unternehmen liegt. Stattdessen verschiebt sich der Fokus hin zur Fertigung in Losgröße 1, das heißt vollständig nach kundenindividuellen Anforderungen und Vorlieben [Ba17; KN09]. Um diesem Bedarf Rechnung zu tragen und weiterhin eine effiziente und automatisierte Produktion zu ermöglichen, müssen zukünftige Produktionssysteme wandlungsfähig und modular gestaltet sein [HB14; KWH13; Zü16]. Produktionssysteme werden also nicht mehr entwickelt, um für lange Produktionsphasen Bestand zu haben, sondern um zeitsparend und kostengünstig an veränderte Anforderungen angepasst und umgerüstet werden zu können.

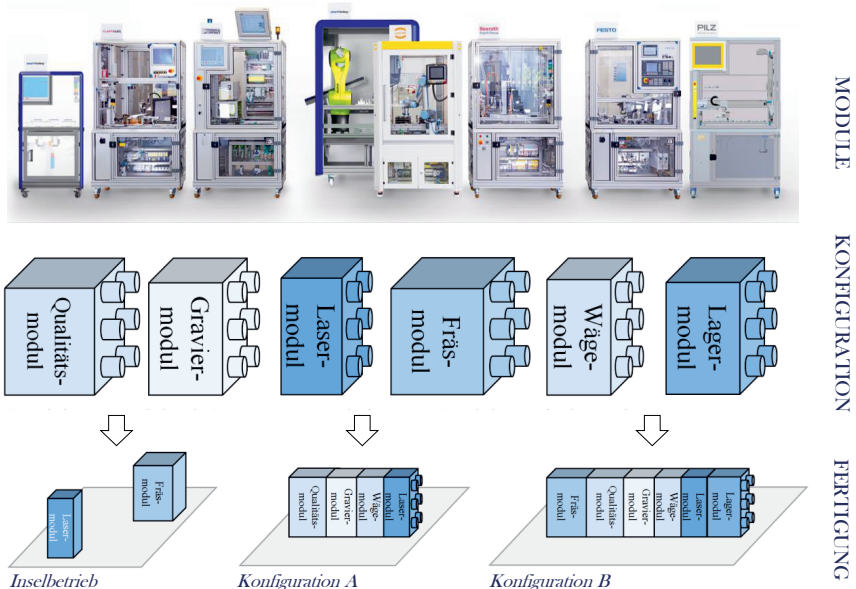


Abbildung 2.11: Modulare und herstellerunabhängige Industrie 4.0-Produktionsanlage nach [We17b]

Die *SmartFactory<sup>KL</sup>* liefert mit ihrem Demonstrator ein Beispiel, wie eine wandlungsfähige und modulare Produktionsanlage aussehen kann (vgl. Abbildung 2.11). Die Anlage besteht aus funktional gekapselten Modulen, die unterschiedliche Fähigkeiten und Bearbeitungstätigkeiten

bereitstellen. Durch die Umsetzung einer vereinheitlichten Schnittstelle für die Energie- und Medienversorgung (Strom, Druckluft und Kommunikation) wurde ein herstellerunabhängiges System nach den Prinzipien eines Baukastens realisiert [Go16; We17a; We17b]. Die Produktionsmodule lassen sich zu verschiedenen Konfigurationen kombinieren, um die Anforderungen des aktuellen Fertigungsauftrages bestmöglich umsetzen zu können.

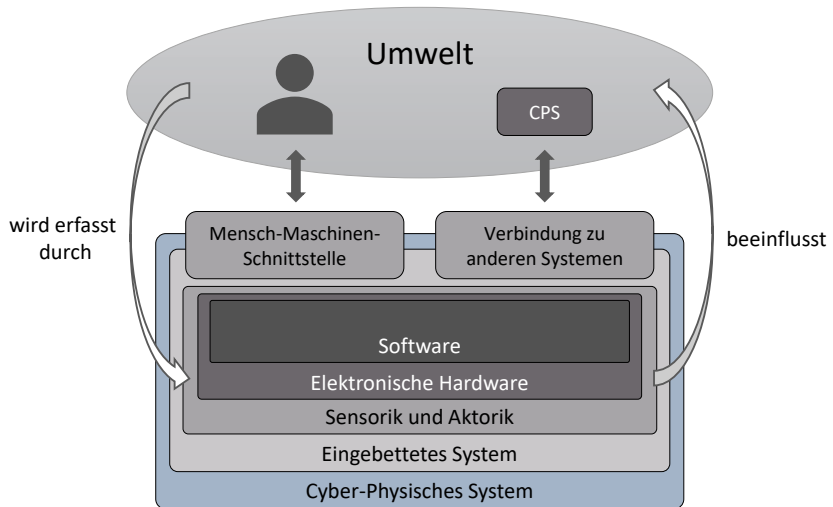


Abbildung 2.12: Schematischer Aufbau eines CPS in Anlehnung an [Br11] und [WT17]

Als Grundlage für den Aufbau von modularen und dezentral vernetzten Systemen sowie wandlungsfähigen Anlagen werden CPS gesehen. Das Merkmal von CPSn ist eine Verknüpfung realer (physischer) Objekte und Prozesse mit informationsverarbeitenden (virtuellen) Objekten über offene und jederzeit miteinander verbundene Informationsnetze [GB12]. CPS bestehen dementsprechend aus elektromechanischen Bestandteilen (Aktoren, Sensoren), Systemen zur Informations- und Datenverarbeitung sowie einer Kommunikationsschnittstelle [Br11]. Durch die physischen Bestandteile können CPS Informationen über ihre Umwelt sammeln, diese Informationen verarbeiten und direkt oder indirekt mit der Umwelt interagieren. Der Aufbau von CPSn entspricht weitestgehend dem Aufbau von eingebetteten Systemen. Der Hauptunterschied zwischen CPSn und eingebetteten Systemen liegt in der offenen Schnittstelle, über die eine Interaktion mit weiteren CPSn oder mit dem Menschen erfolgt. Hierdurch ergibt sich ein wesentlicher Unterschied in der Einsatzweise von CPSn. Abbildung 2.12 zeigt den schematischen Aufbau eines CPS in Anlehnung an die Darstellung von BROU, die mit Elementen nach WIESNER UND THOBEN erweitert wurde [WT17]. Die offenen Schnittstellen und die Fähigkeit der Informationsverarbeitung ermöglicht CPSn eine in großen Teilen autonome Arbeitsweise im Produktionsumfeld. Die eigene Logik und Intelligenz von CPSn trägt dazu bei, fehlertolerante Gesamtsysteme aufzubauen [Ba17]. Die Interoperabilität der Systeme und der CPS zählt daher

als wichtiges Paradigma in der Industrie 4.0 [GB12]. Werden eine Vielzahl von CPSn in diesem Sinne kombiniert, spricht man auch von CPPM [Bi18] oder gar von CPPS.

Der Trend hin zu wandlungsfähigen und modularen Anlagen birgt jedoch auch Herausforderungen für die Betreiber und die Bediener solcher Produktionsanlagen. Durch die Wandlungsfähigkeit des Produktionsumfelds ist es schwieriger, den Überblick über den aktuellen Aufbau der Anlage, die Aufgaben von Produktionsmodulen und auftretende Probleme zu behalten. Als Konsequenz dieser Entwicklung rücken menschenzentrierte und anpassbare Benutzungsschnittstellen in den Fokus der Forschung und Wissenschaft, um eine einfache Interaktion mit den modularen Produktionsanlagen zu ermöglichen [Ab17; Bi18; BMR19]. Immer mehr Unternehmen sehen daher die Vorzüge von mobilen und dezentral vernetzten Assistenzsystemen, die mit den verteilten, CPS-basierten Produktionsanlagen interagieren. Durch die CPS bereitgestellte Informationen können durch Assistenzsysteme analysiert sowie situations- und nutzergerecht dargestellt werden [BQR17; Sp13]. Um den vielfältigen und sich veränderten Anforderungen gerecht zu werden, die sich für Assistenzsysteme in einem wandlungsfähigen Produktionsumfeld ergeben, werden Assistenzsysteme zunehmend nach denselben Prinzipien der Modularisierung und durch die Kombination von CPSn aufgebaut [Mü18a; Si19]. Diese Entwicklung zeigt sich sowohl im Forschungskontext als auch im industriellen Umfeld, indem sich erste modulare Assistenzsysteme finden, die erweitert und an die Anforderungen der Situation angepasst werden können. Als Beispiel sei hierbei auf das ActiveAssist-System für Handarbeitsplätze in der variantenreichen Montage verwiesen [Rexroth 1]. Die Grundvariante des Systems beinhaltet eine interaktive, displaybasierte Unterstützung durch Arbeitsanweisungen. Erweitert werden kann das System beispielsweise durch Projektionslösungen zur visuellen Unterstützung (vgl. Abschnitt 2.1.2), Handerkennungsmodulen zur Kontexterfassung (vgl. Abschnitt 2.1.2 und Abschnitt 2.1.3) oder einer Schrauberintegration zur Steuerung und Überwachung von Schraubprozessen, die mit fest definierten Drehmomenten durchgeführt werden müssen.

Der Bedarf an modularen und anpassbaren Assistenzsystemen zeigt sich vor allem im Bereich der Reparatur und Nacharbeit von kundenindividuellen Produkten, die in kleinen Losgrößen bis hin zu Losgröße 1 gefertigt werden [Mü18a; Mü18b]. Im Gegensatz zur reinen Montage, bei der es angepasste Arbeitsplätze für jedes Produkt gibt, sind Arbeitsplätze in der Nacharbeit und Reparatur durch ihren flexiblen und universellen Aufbau gekennzeichnet. Durch den universellen Aufbau können Reparaturarbeiten für ein breites Spektrum an Produkten realisiert werden. Ein Bedarf an spezifische Reparaturplätze für jede Situation besteht daher nicht. Der anpassbare Aufbau dieser Arbeitsplätze bedingt jedoch ebenfalls einen flexiblen Aufbau der Assistenzsysteme für diese Arbeitsplätze, zum Beispiel in Form von modularen Systemen.

Eine weitere Herausforderung im Bereich der Reparatur und Nacharbeit ist es, dass der Ursprung eines Defektes und somit die notwendigen Schritte zur Behebung häufig nicht von vornherein bekannt sind. Aus diesem Grund kann vom Assistenzsystem kein optimaler Arbeitsweg durch alle Arbeitsschritte vorausbestimmt werden. Stattdessen bedarf es der Flexibilität und Re-

aktionsfähigkeit des Menschen [TS02]. Reparaturaufgaben werden daher von speziell ausgebildeten Fachkräften mit langjähriger Erfahrung durchgeführt, die sich anhand von Indizien und ihrem Erfahrungswissen der Problemlösung annähern und eine Reparatur durchführen [Mü18a]. Gerade beim hohen Variantenreichtum und den unterschiedlichen Montageprozessen bei einer Fertigung nach Losgröße 1 verlieren die Facharbeiter jedoch zunehmend den Überblick über die notwendigen Handgriffe, um eine schnelle und fehlerfreie Reparatur durchführen zu können. Assistenzsysteme für die Reparatur und Nacharbeit von Losgröße 1-Produkten müssen daher flexibel genug sein, um sich einerseits an Veränderungen und an den Arbeitsweg der Fachkraft anzupassen und andererseits eine Unterstützung bei der Durchführung von Reparaturprozessen zu bieten [Ba08; LZB17; Mü16].

### 2.1.5 Zusammenfassung und Abgrenzung

Es existiert eine große Auswahl an Assistenzsystemen im Bereich der Wissenschaft und Forschung sowie im industriellen Umfeld, die den Menschen bei seiner Arbeit im Produktionsumfeld unterstützen [Be18b]. Aufgrund der großen Bandbreite an Zielsetzungen legen die Systeme unterschiedliche Schwerpunkte bezüglich ihrer Unterstützungsfähigkeit [Ap18]. Die Bandbreite reicht von Systemen, die sich auf die richtige Informationsdarstellung zur Reduzierung von Trainingszeiten spezialisieren [SG15] bis hin zu Systemen die automatisch Assistenzinformationen aus anderen Wissensquellen generieren und bereitstellen [AB15]. HINRICHSSEN UND BORNEWASSER beschreiben zudem nach welchen Prinzipien und Empfehlungen Assistenzsysteme aufgebaut werden sollen [HB19]. Ein zentrales Kernelement aller interaktiven Systeme ist die kontextsensitive Reaktion auf die arbeitende Person und / oder den Montageprozess.

In Abgrenzung zu den präsentierten Systemen beschäftigt sich diese Arbeit mit der Realisierung eines kontextsensitiven, modularen Assistenzsystems im Bereich der Reparatur und Nacharbeit von Produkten. Der Fokus der Arbeit liegt auf der Umsetzung eines flexiblen Systems, das sich an die Arbeitswege der nutzenden Person anpasst, gleichzeitig aber die kausalen Zusammenhänge zwischen den Arbeitsschritten des Arbeitsprozesses beachtet.

Verwandte Arbeiten finden sich im Rahmen des Plant@Hand-Systems zur Aktivitätserkennung [AGU14] und Erfassung von Arbeitsprozessen [BA14], um eine situationsabhängige und nutzerzentrierte Informationsassistenz umzusetzen. Die Arbeiten beschäftigen sich mit flexiblen Arbeitsplänen und der Bereitstellung von Informationen zu dem Arbeitsschritt, der am wahrscheinlichsten durchgeführt wird. Das System ist jedoch für die manuelle Montage vorgesehen, bei der der zu erreichende Endzustand bereits bekannt ist, wodurch zu jedem Zeitpunkt eine ideale Montagesequenz berechnet und der nutzenden Person eine darauf abgestimmte Handlungsempfehlung angeboten werden kann. Darüber hinaus ist das Plant@Hand-System nicht durch einen modularen Aufbau geprägt.

WÖLFLE beschäftigt sich in seinen Arbeiten mit der technischen Ausgestaltung eines kontextsensitiven Assistenzsystems im Bereich der Intralogistik [Wö14]. Der Fokus der Arbeit liegt vor allem an der Auswahl und Kombination von technischen Systemen und Technologien sowie deren Nutzung, um ein kontextsensitives System zu realisieren. Aufgrund der Flexibilität von Logistikprozessen berücksichtigt WÖLFLE flexible Bearbeitungsreihenfolgen. Jedoch stehen weder die flexible Bearbeitung durch die nutzende Person im Mittelpunkt, noch werden Arbeitsprozesse mit kausal zusammenhängenden Arbeitsschritten betrachtet. Die Realisierung des Systems folgt einem festen Aufbau und ist nicht durch einen modularen CPS-basierten Ansatz geprägt.

Die Arbeiten von KORN und FUNK beschäftigen sich mit kontextsensitiven Assistenzsystemen (CAAS), die sich flexibel an die Arbeitsweise und die Probleme der nutzenden Person anpassen. Die Fokusse liegen auf der richtigen Informationsdarstellung und der Erfassung der mentalen Arbeitsbelastung, um die nutzenden Personen zu fordern und zu animieren, aber nicht zu überlasten. Die Umsetzung eines Systems zur flexiblen Bestimmung des gewählten Arbeitsweges und ein modularer Aufbau werden nicht weiter betrachtet.

Zudem finden sich Arbeiten im Bereich der Nacharbeit und Reparatur technischer Systeme ohne fest vorgegebene Arbeitsweise und Reihenfolge [Mü18a; Mü18b] sowie Arbeiten die sich mit Process-Mining beschäftigen [Kn17; Kn18], das heißt der flexiblen Erfassung von nicht fest definierten Arbeitsschritten. Zusätzlich betrachten MÜLLER ET AL. modulare / flexible Assistenzsysteme. Der Kern der Arbeiten liegt jedoch auf der Planung und flexiblen Anpassung der Arbeitspläne an bekannte Reparaturaufträge. Eine flexible und kontextsensitive Anpassung auf andere Arbeitswege während des Arbeitsprozesses werden nicht tiefergehend untersucht. KNOCH ET AL. hingegen beschäftigen sich mit der flexiblen Ausführung und kontextsensitiven Erfassung von Arbeitsschritten, jedoch liegt dort kein Fokus auf modularen, CPS-basierten Assistenzsystemen.

## 2.2 Grundlagen zur semantischen Modellierung des Arbeitskontexts

Die semantische Modellierung des Arbeitskontexts ist ein essentieller Bestandteil im Rahmen dieser Arbeit, um eine kontextsensitive Erfassung der aktuellen Arbeitssituation und des gerade durchgeführten Arbeitsschritts zu ermöglichen. Im Verlauf dieses Kapitel werden daher zuerst relevante Begrifflichkeiten sowie allgemeine Kontextmodelle vorgestellt (2.2.1). Anschließend erfolgt eine Fokussierung auf spezifische Konzepte des Produktionsumfeldes (2.2.2) gefolgt von Informationsmodellen zu manuellen Tätigkeiten (2.2.3) und zu Werkzeugen und Materialien (2.2.4) als Grundlage zur praktischen Erstellung eines semantischen Kontextinformationsmodells. Den Abschluss dieses Kapitels bildet eine Zusammenfassung (2.2.5).



### 2.2.1 Kontextmodellierung: Begrifflichkeiten und bekannte Kontextmodelle

Die Bedeutung von Kontext und was wir unter Kontext verstehen ist für die meisten Menschen aufgrund von eigenen Erfahrungen intuitiv und implizit bekannt [De01]. Bei jedem Gespräch und jeder Interaktion wird der Kontext der Situation bewertet und fließt in das Verhalten und die eigene Reaktion mit ein. Soll die Bedeutung von Kontext jedoch erklärt werden, fällt es schwer, eine eindeutige Definition zu geben. Insbesondere für die Mensch-Maschinen-Interaktion ist eine Definition des Kontexts jedoch von großer Bedeutung. Im Gegenteil zu Menschen, die es intuitiv verstehen, den Kontext zu bewerten und in das Verhalten einfließen zu lassen, besitzen Computersysteme diese Fähigkeit nicht. Relevante Informationen für die Interaktion mit einem Computersystem müssen entweder explizit eingegeben werden oder das System muss in die Lage versetzt werden, den relevanten Kontext selbst zu erfassen. Aus diesem Grund beschäftigen sich mehrere Arbeiten damit eine Definition von Kontext zu bestimmen. Beispiele hierfür finden sich in [Br95; Pa98; Ro98; ST94; WJH97]. Eine weitverbreitete Definition von ABOWD UND DEY beschreibt Kontext als die Menge aller Informationen, die verwendet werden können, um die Situation einer Entität zu charakterisieren. Als Entität werden bei dieser Definition Personen, Orte oder Objekte angesehen (vgl. Kontextdefinition nach [Ab99; De01]).

**Kontextdefinition nach [Ab99; De01]:**

Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.

Im Gegensatz zu den anderen Definitionen zielt DEY auf eine allgemeingültige Beschreibung des Kontexts, die sich nicht auf ein spezifisches Szenario oder eine Anwendung bezieht. Entwickeln von kontextsensitiven Systemen soll die Definition helfen, ein passendes System für das eigene Szenario umzusetzen, ohne dabei durch eine allzu detaillierte Beschreibung eines spezifischen Beispiels eingeschränkt zu werden.

Nach ABOWD UND DEY ist ein System dann kontextsensitiv, wenn es den Kontext einer Situation erfasst und dafür verwendet, der nutzenden Person relevante Informationen oder Funktionen bereitzustellen [Ab99]. Die Relevanz der Informationen und Funktionen richtet sich nach der Aufgabe, welche die nutzende Person zum entsprechenden Zeitpunkt durchführt.

**Definition von Kontextsensitivität nach [Ab99; De01]:**

A system is context-aware if it uses context to provide relevant information and / or services to the user, where relevancy depends on the user's task.

Die Nutzung von kontextsensitiven Systemen findet sich beispielsweise bei Indoor-Shoppingassistenten [OP17], die Einkaufshinweise auf Basis einer nutzerindividuellen Wunschliste geben, oder bei elektronischen Touristenführern [Ch00], die in Abhängigkeit von den persönlichen Interessen und der Umgebung Informationen über Sehenswürdigkeiten

bereitstellen. Im Rahmen dieser Arbeit sind insbesondere Assistenzsysteme für die manuelle Montage und die Reparatur von technischen Systemen Beispiele für kontextsensitive Systeme (vgl. Abschnitt 2.2.3).

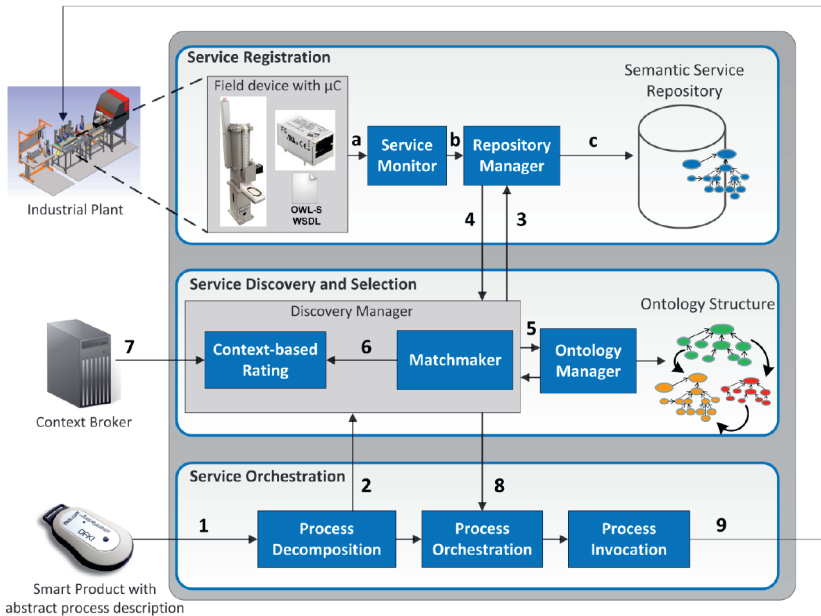


Abbildung 2.13: Kontextbasierte Orchestrierung von Feldgerätefunktionalitäten (Quelle: [Lo13])

Bei der Definition nach ABOWD UND DEY ist der direkte Bezug zwischen Kontextsensitivität und der Aufgabe der nutzenden Personen auffällig, wodurch lediglich Interaktionen zwischen Mensch und Maschine berücksichtigt werden. Aus diesem Grund erweitert LOSKYLL ET AL. die Definition dahingehend, dass sich die Relevanz von Informationen nicht nur auf die Aufgabe einer nutzenden Person, sondern allgemein auf die Aufgabe einer beliebigen Entität bezieht [Lo12]. Ein Beispiel für diesen Fall ist die autonome, modulare Produktion, in der eine Orchestrierung von verfügbaren Arbeitsstationen in Abhängigkeit des abzuwickelnden Arbeitsauftrages und von den Fähigkeiten der verfügbaren Arbeitsstationen durchgeführt wird (vgl. Abbildung 2.13) [Lo13; Pf15; Wa14].

Nach ZIMMERMANN ET AL. zeigt sich eine weitere Auffälligkeit der Kontextdefinition in der Verwendung unbestimmter Ausdrücke, wie „jede Information“ und „die Situation charakterisiert“ [ZLO07]. Zwar erreicht DEY mit der Verwendung dieser Begriffe eine generalisierte Definition von Kontext, jedoch fehlt es dadurch an Einschränkungen, wodurch die praktische Nutzung und die einfache Erklärbarkeit für Anwender erschwert wird [ZLO07]. Um eine bessere Nutzbarkeit zu erreichen, beschäftigen sich unter anderem CHEN, CROWLEY, KOFOD-PETERSEN

UND HENRICKSEN mit der Arbeitsaufgabe einer Entität als zentrales Element bei der Definition des Kontexts [Ch04; Cr02; He03; KC05]. Insbesondere deren Annahme, dass die Durchführung der Arbeitsaufgabe immer auf ein Ziel ausgerichtet ist, zeigt die Wichtigkeit der Arbeitsaufgabe für die Kontextsensitivität von unterstützenden Systemen. Aus diesem Grund führen CROWLEY UND KOFOED-PETERSEN den Ausdruck der „Aktivität“ als explizite Erweiterung zur Beschreibung der Arbeitsaufgabe ein [Cr02; KC05].

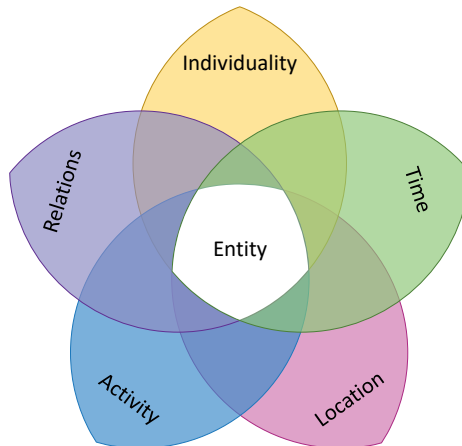


Abbildung 2.14: Fünf fundamentale Kategorien von Kontextinformationen (Quelle: [ZLO07])

Darüber hinaus erweitert ZIMMERMANN ET AL. die Definition von DEY um einen formellen und einen operativen Teil, um diese einfacher nutzbar und verständlich zu machen [ZLO07]. Im formalen Anteil werden die fünf Kategorien „Individualität“, „Zeit“, „Ort“, „Aktivität“ und „Relation“ zur Einordnung von Kontextinformationen ergänzt (vgl. Abbildung 2.14). Zur akkuraten Beschreibung des Kontexts einer Entität sind demnach Kontextinformationen aus allen Kategorien erforderlich. Jedoch verschiebt sich je nach Situation der Fokus des Kontexts auf eine oder mehrere Kategorien.

Die Kategorie der „Individualität“ umfasst alle Kontextinformationen, die die zentrale Entität des Kontexts näher beschreiben oder einordnen. Hierzu zählen beispielsweise der Zustand der Entität oder die Art der Entität, um die es sich handelt. ZIMMERMANN unterscheidet in seinem Modell menschliche Entitäten, natürliche Entitäten (wie zum Beispiel Pflanzen oder Steine), künstliche Entitäten und die Kombination von Entitäten als Gruppe [ZLO07]. Unter der Kategorie „Zeit“ werden Kontextinformationen zusammengefasst, die temporale Abhängigkeiten des Kontexts oder der Entität beschreiben. Beispiele hierfür sind die Zeitzone der Entität oder die aktuelle Uhrzeit. Aber auch historische Informationen über Kontextinformationen oder Informationen über sich wiederholende Ereignisse werden in dieser Kategorie erfasst.

Ähnlich wie temporale Informationen sind örtliche Informationen wichtig, um die Situation und dessen Kontext näher zu beschreiben. Aus diesem Grund werden alle Kontextinformationen, die dem physischen oder auch virtuellen Einordnen eines Ortes dienen unter der Kategorie „Ort“ zusammengefasst. Hierzu zählen geographische Standortdaten zum Beispiel von Ortungssystemen wie GPS (Global Positioning System), quantitative Informationen (wie Gebäude oder Straßen) aber auch relative Positionen zu anderen Entitäten oder IP-Adressen (Internet Protokoll) zur Bestimmung des Ortes in einem Netzwerk. Zur Kategorie „Aktivität“ gehörende Kontextinformationen beschreiben die Ziele und Aufgaben einer Entität und geben Auskunft darüber, was die Entität gerade tut oder was sie in Zukunft tun wird. Im Bezug zu kontextsensitiven Systemen wird der Aktivitäten-Kontext häufig durch domänenspezifische Aufgaben und Unteraufgaben, als Ziel einer Gesamtaufgabe, repräsentiert.

Die letzte Kategorie umfasst alle Kontextinformationen, die die „Relationen“ einer Entität zu anderen Entitäten abbildennet. Entsprechend der vorherigen Beschreibung können andere Entitäten wiederum in menschliche, künstliche und natürliche Entitäten sowie Gruppen von Entitäten eingeordnet werden. In Abhängigkeit von den beteiligten Entitäten unterscheidet ZIMMERMANN zwischen sozialen, funktionalen und kompositionellen / assoziativen Relationen [ZLO07]. Zu den sozialen Relationen zählen demnach Kontextinformationen zu zwischenmenschlichen Beziehungen, wie sie zum Beispiel zwischen Freunden, Familienmitgliedern oder Arbeitskollegen vorherrschen. Unter den funktionalen Relationen werden Informationen über die Nutzung einer anderen Entität zur Erreichung eines Zieles zusammengefasst. Hierzu gehört beispielsweise die Nutzung eines Schraubendrehers, um eine Schraube herauszudrehen. Kompositionelle / assoziative Relationen beschreiben hingegen die Verbindungen zwischen einer Gesamtentität und ihren Einzelteilen. Ein Beispiel hierfür ist ein fertiges Produkt, das aus Baugruppen besteht.

## 2.2.2 Kontextmodelle für das Produktionsumfeld

Zusätzlich zu allgemeinen Modellen zur Beschreibung und Abbildung des Kontexts existieren spezifische Kontextmodelle für die Produktionsdomäne. ALM UND HADLAK präsentieren ein solches Modell zur Integration einer kontextbasierten Wissensbereitstellung in eine Aktivitätserkennung für Fertigungsaufträge in der manuellen Montage [AH15]. Grundlage zur Herleitung des Modells ist eine Umfrage zur Bestimmung der allgemeinen Anforderungen, die an arbeitsbezogene Informationen gestellt werden, und der Faktoren, die die Bereitschaft ein Assistenzsystem zu nutzen beeinflussen. Die Ergebnisse der Umfrage zeigen zum Beispiel, dass die Bereitstellung von Informationen zu den verwendeten Werkzeugen (93,54 %) und zu den zu verwendenden Materialien (100 %) wichtig ist.

In Anlehnung an DEY wurden auf Basis der Ergebnisse vier kontextrelevante Kategorien für die manuelle Montage in der Produktionsdomäne abgeleitet. Die Modellierung dieser Kategorien

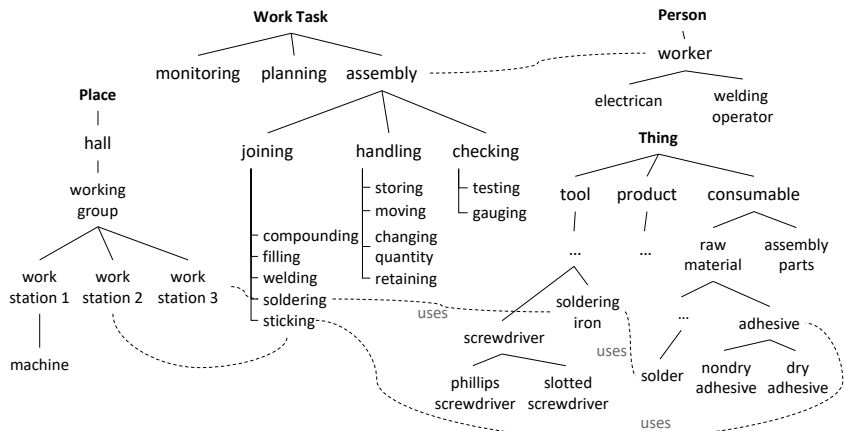


Abbildung 2.15: Beispielontologie zum kontextuelle Modell von Alm und Hadlak (Quelle: [AH15])

und deren Verbindungen untereinander erfolgte mithilfe einer Ontologie. Abbildung 2.15 zeigt eine Beispielontologie mit den vier Kategorien *Person*, *Ort*, *Gegenstand / Ding* und *Arbeitsaufgabe*. Die Kategorien sind in einer hierarchischen Form, vom Groben ins Detail gehend, weiter unterteilt (durchgehende Linien). Dementsprechend kann eine Aufgabe das Überwachen oder aber die Durchführung eines Fertigungsschritts sein. Der Fertigungsschritt selbst kann wiederum unterteilt werden in zum Beispiel Handhabungstätigkeiten oder Montagetätigkeiten. Zusätzlich bestehen kontextuelle Beziehungen zwischen spezifischen Elementen der verschiedenen Kategorien (gestrichelte Linien). Zentrales Element ist die Arbeitsaufgabe (*Work Task*), die von einer arbeitenden Person (*Person*) mit Werkzeugen (*Tools*) an einem Arbeitsplatz (*Place*) durchgeführt wird. Bei diesem Prozess werden Materialien und Teile (*Consumables*) verbraucht, um ein Produkt (*Product*) herzustellen.

Ein weiteres Modell zur Beschreibung des Kontexts im Produktionsumfeld stammt von CLAEYS ET AL.. Aufbauend auf dem Modell von ALM UND HADLAK wird das Modell um die vier Kontextkategorien *Erfahrung*, *Umgebungszustand*, *Verwendung* und *Anforderungen* erweitert, um die kontextuellen Beziehungen zwischen den Kategorien von ALM UND HADLAK explizit modellieren zu können [CI18]. So beschreiben Kontextinformationen der Kategorie *Erfahrung* die Fähigkeiten und Fertigkeiten einer Person, eine bestimmte Aufgabe zu bearbeiten. Indikatoren hierfür sind zum Beispiel die Durchlaufzeit, der letzte Zeitpunkt der Durchführung oder die gemachten Fehler bei der Bearbeitung der Aufgaben. In die Kategorie *Anforderungen* fallen Kontextinformationen, die die notwendigen Gegenstände / Materialien (*Thing*) und den Ort (*Place*) beschreiben, um eine spezifische Aufgabe durch eine Person durchführen lassen zu können. Umgebungsfaktoren des Produktionsprozesses, wie die Temperatur für einen Schweißprozess oder die Luftfeuchtigkeit, werden unter der Kategorie *Umgebungszustand* zusammengefasst.

Kontextinformationen der Kategorie *Anwendung* sind eng mit Informationen der Kategorie *Anforderungen* verknüpft. Während Kontextinformationen der Kategorie *Anforderungen* jedoch lediglich die notwendigen Anforderungen bezüglich einer Arbeitsaufgabe beschreiben, modelliert der Anwendungs-Kontext die direkten Beziehungen zwischen Dingen und Orten. Dies beinhaltet Informationen darüber, wie ein spezielles Werkzeug an einem bestimmten Ort zu verwenden ist. Abbildung 2.16 zeigt das kontextuelle Abhängigkeitsmodell von [CI18].

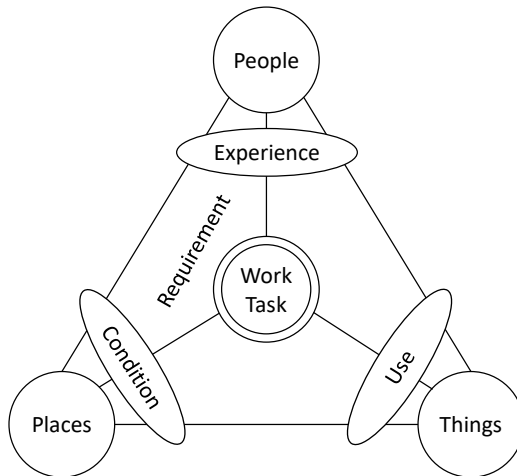


Abbildung 2.16: Kontextuelles Abhängigkeitsmodell für Fertigungsaufgaben nach [CI18]

Das Ziel des Ontologie-Modells von CLAEYS ET AL. ist das kontextsensitive Verwalten und Beschreiben von Arbeitsanweisungen, um einerseits die Durchführung von Arbeitsanweisungen automatisiert zu erfassen und andererseits eine Filterung verfügbarer Arbeitsanweisungen, auf Basis der automatisierten Erfassung, zur situationsangepassten Anzeige von Unterstützungsleistungen durchzuführen [CI18]. Im Gegensatz zum Modell von ALM UND HADLAK ist dieses Modell allgemeingültiger und weniger auf einen spezifischen Aufgabentyp fokussiert, weshalb keine weitere Untergliederung der Kategorien erfolgt.

Ähnlich zu dem Modell von ALM UND HADLAK präsentieren AEHNELT ET AL. ein produktionsbezogenes Kontextmodell für die manuelle Montage [AGU14]. Das Modell unterscheidet zwischen *Orten*, *Personen* und *Gegenständen / Dingen* (vgl. Abbildung 2.17). Im Gegensatz zu ALM UND HADLAK verzichten AEHNELT ET AL. auf eine eigenständige Kategorie für die Arbeitsaufgabe. Stattdessen wird die Arbeitsaufgabe mit ihren einzelnen Arbeitsschritten der Kategorie der *Dinge* zugeordnet. Zudem umfasst diese Kategorie Kontextinformationen zu Werkzeugen, Materialien, Bauteilen / -komponenten und Produkten. Die Arbeitsschritte definieren hierbei, welche Werkzeuge und welche Materialien oder Bauteile notwendig sind und wie diese zu verwenden sind, um ein Produkt zu montieren. Die Handhabung von Werkzeugen und Gegenständen sowie die Bearbeitung der Arbeitsaufgabe wird entsprechend dem Modell durch eine Person

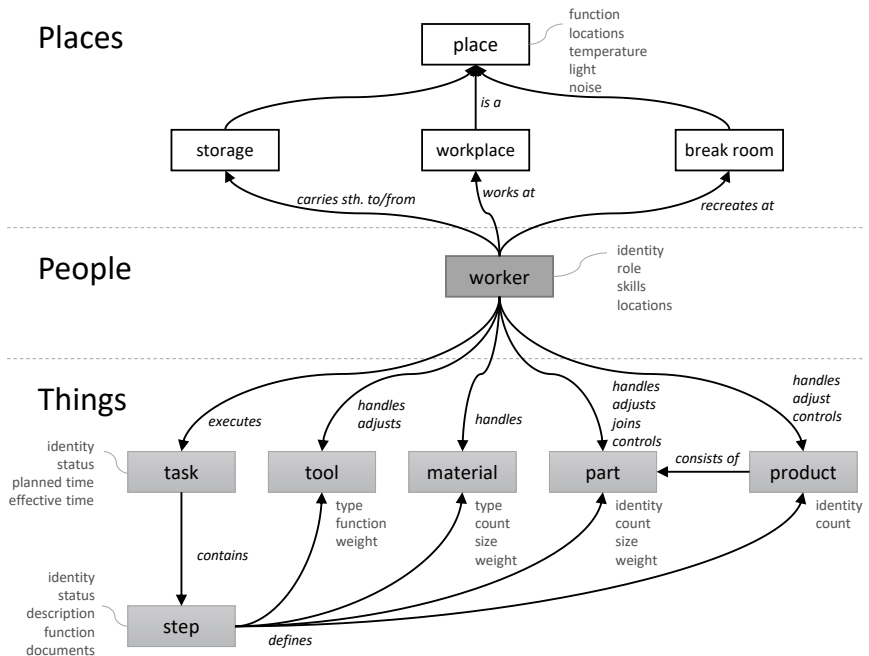


Abbildung 2.17: Fertigungs-Kontextmodell für die manuelle Montage (Quelle: [AGU14])

durchgeführt. Aufgrund der Ausrichtung auf die manuelle Montage fokussiert das Modell eine *arbeitende Person* als spezifisches Element der Kategorie *Person*. Darüber hinaus werden das *Lager*, der *Arbeitsplatz* und der *Pausenraum* als Ausprägungen des *Orts* definiert, wobei mit jedem dieser Orte individuelle Tätigkeiten assoziiert werden. So bringt oder holt die *arbeitende Person* etwas aus oder vom *Lager*, während *Arbeitsaufgaben* am *Arbeitsplatz* verrichtet werden und eine Erholung im *Pausenraum* erfolgt.

### 2.2.3 Modelle zur Beschreibung manueller Tätigkeiten im Produktionsumfeld

Wird nicht nur die abstrakte Ebene des vollumfänglichen Arbeitskontexts betrachtet, sondern der Fokus auf die eigentliche Aktivität der Arbeit gelegt, finden sich weitere Möglichkeiten zur Kontextmodellierung. So beschreibt die VDI 2860 die Montage als Kombination aller Aktivitäten und Vorgänge, die für das Zusammenbauen von Körpern notwendig sind. Dementsprechend umfasst die Montage *Fertigungsverfahren* der DIN 8580, *Handhabungsprozesse* der VDI 2860 sowie weitere Sondertätigkeiten als Unterstützung des Montageprozesses [LW13]. Unter Berücksichtigung des Anwendungsfalles der Reparatur von technischen Geräten, bei der eine De-

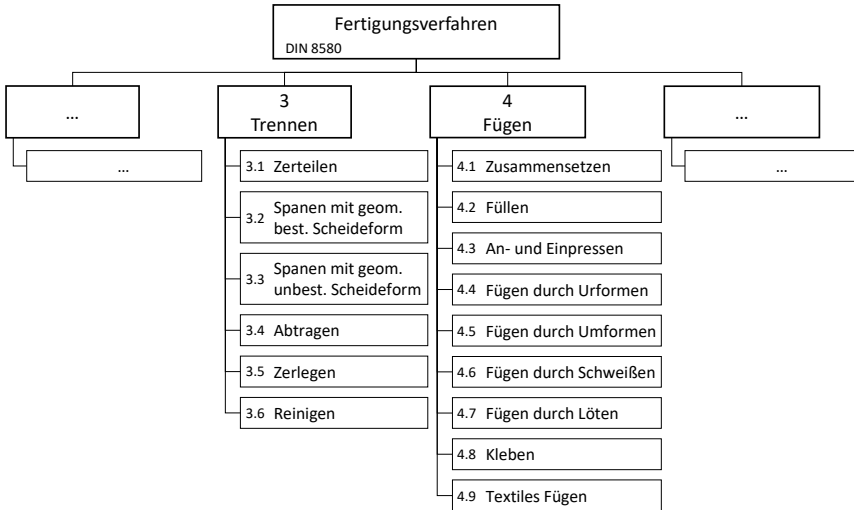


Abbildung 2.18: Ausschnitt der Fertigungsverfahren Trennen und Fügen nach [DIN 8580]

montage und anschließende Remontage durchgeführt werden, sind insbesondere die Fertigungsverfahren des *Fügens* und des *Trennens* von Bedeutung. *Fügen* beschreibt dabei das Zusammenbringen zweier oder mehrerer Körpern, beispielsweise durch *Zusammensetzen* oder *Verkleben*. Im Gegenzug zum *Fügen* beschreibt *Trennen* das Auseinanderbringen zweier oder mehreren Körpern beispielsweise durch *Zerteilen* oder eine mechanische Bearbeitung. Abbildung 2.18 zeigt einen Ausschnitt der *Fertigungsverfahren* nach DIN 8580, in dem alle Operationen der Fertigungsverfahren des *Trennens* und des *Fügens* dargestellt sind.

Während Fertigungsverfahren eine Veränderung am Produkt bewirken, gehören Handhabungsaufgaben zu Nebentätigkeiten, um Fertigungsoperationen durchführen zu können. Die VDI-Richtlinie 2860 unterteilt die Handhabung dabei in die fünf Teilfunktionen *Speichern*, *Mengen verändern*, *Bewegen*, *Sichern* und *Kontrollieren*, wobei sich die Teilfunktionen wiederum in Unterfunktionen aufteilen (vgl. Abbildung 2.19). Die Unterfunktionen lassen sich dabei in die Gruppe der *elementaren Funktionen* und der *zusammengesetzten Funktionen* einordnen, wobei *elementare Funktionen* die kleinsten nicht weiter unterteilbaren Funktionen einer Kategorie darstellen. *Zusammengesetzte Funktionen* hingegen stellen eine Kombination aus zwei oder mehreren *elementaren Funktionen* dar.

Bezüglich der VDI-Norm 2860 ist jedoch anzumerken, dass die zur Zeit dieser Arbeit neueste Version aus dem Jahr 1990 stammt und im Jahr 2018 zurückgezogen wurde. Der Rückzug der Norm erfolgte unter dem Hinweis, dass diese nicht mehr dem aktuellen Stand der Technik genügt und eine Überarbeitung notwendig ist [VDI]. Zwar ist eine Neufassung für das Jahr 2020 geplant, nach aktuellem Kenntnisstand liegt bis dato jedoch keine alternative Quelle für die Beschreibung



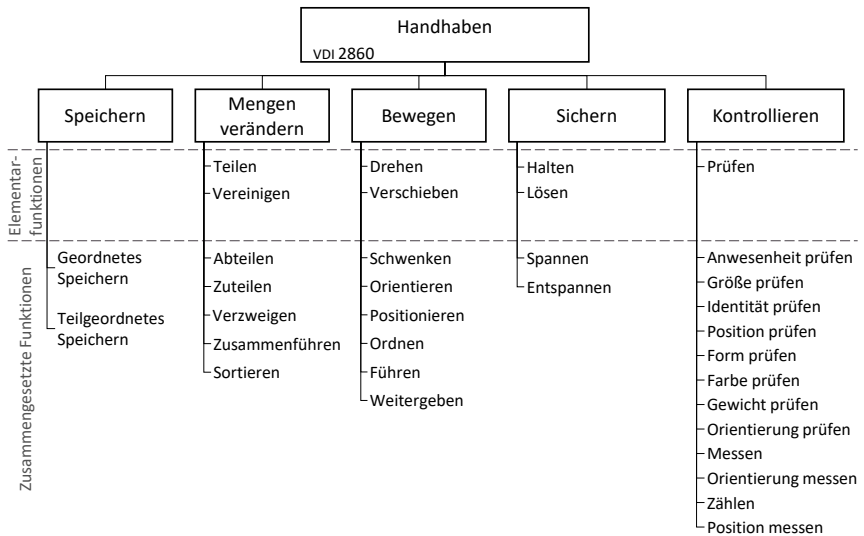


Abbildung 2.19: Teilfunktionen des Handhabens nach [VDI 2860]

von Montage- und Handhabungstechniken vor, sodass weiterhin die zurückgezogene Norm als aktueller Stand der Technik angesehen wird.

Die Nutzung von menschlichen Tätigkeiten und Aufgaben als Informationsquelle und zur Modellierung der Fähigkeiten eines Produktionsumfelds nimmt weiterhin zu. Beispiele für die Nutzung finden sich unter anderem in [AH15; Cl18; Fe17]. Für die Betrachtung innerhalb der Arbeiten dieser Autoren stellt die Modellierung und Klassifizierung der Arbeitsschritte keine Schwerpunkt, sodass diese entweder nicht weiter betrachtet werden (vgl. [AGU14; AH15; Fe17]) oder auf die Prozesse und Operationen der VDI 2860 und DIN 8580 verwiesen werden (vgl. [AGU14]). Trotz der Nachfrage nach einer einheitlichen Beschreibung und Modellierung von manuellen Tätigkeiten in der Montage [Be17] existiert bis dato noch keine allgemeingültige Formalisierung zur Klassifikation dieser Tätigkeiten. Jedoch finden sich erste Arbeiten, die sich mit der Umsetzung einer einheitlichen Sprache für manuelle Tätigkeiten im Produktionsumfeld beschäftigen. Aufbauend auf den Elementarfunktionen der VDI 2860 leitet WANSCH zusammengesetzte Tätigkeiten zur Montage und Demontage eines Stellantriebes ab [Wa18]. Hierbei baut er auf den Fertigungsverfahren der DIN 8580 als bekannte zusammengesetzte Operationen auf. Die so gewonnenen manuellen Tätigkeiten werden mit den notwendigen Aufgaben zur Montage und Demontage des Stellantriebes verglichen. Für nicht abgedeckte Aufgaben leitet WANSCH eigene manuelle Tätigkeiten her, indem er Elementarfunktionen der VDI 2860 kombiniert und diese Kombinationen als neue Tätigkeiten erfasst.



Die Grundlage solcher Klassifizierungssysteme bieten Richtlinien und Normen wie die ISO 13584-42 oder die IEC 61360-1, welche die Prinzipien zur Definition von Eigenschaften oder Attributen von Produkten festlegen oder bereits Klassifikationen von Gegenständen und Produkten liefern. Unterschiede in den Klassifikationssystemen finden sich vor allem in deren Fokus. Während *ETIM* sich auf Produktdaten des Fachbereichs der Elektrotechnik sowie damit verwandten Branchen spezialisiert, fokussiert sich *eCI@ss* auf keine spezifische Domäne. Stattdessen bietet *eCI@ss* einen umfänglichen Überblick zu Produkten über alle Branchen und Länder hinweg.

### 2.2.5 Zusammenfassung

Mit den Arbeiten von DEY, ZIMMERMANN ET AL. existieren bereits Modelle, die zur Repräsentation und Modellierung des Kontexts einer Situation verwendet werden können. AEHNELT, CLAEYS UND ALM präsentieren zudem Spezifizierungen und Anpassungen dieser Modelle für die Verwendung von manuellen Tätigkeiten im Fertigungsumfeld. Die Gemeinsamkeit aller Modelle ist die Unterteilung des Kontexts in die Teilbereiche *Ort*, *Gegenstand*, *Aufgabe* und *Person*. Diese Modelle sowie die Klassifizierungen von Teilbereichen des Kontexts bieten eine Grundlage für die Entwicklung eines semantischen Modells und die Separation des Modells in eigenständige Teilmodelle, mit denen die Modellierung des Arbeitskontexts und von Arbeitssituationen bei der Reparatur technischer Systeme umgesetzt werden kann. Für die Bereiche *Aktivität / Aufgabe* und *Gegenstand*, der Modelle von AEHNELT, CLAEYS UND ALM, stehen darüber hinaus weitere Modelle und Informationsmedien bereit. Hierzu gehören die VDI 2860 und die DIN 8580, in Verbindung mit der Arbeit von WANSCH. Diese liefern Informationen zur Modellierung von Tätigkeiten der Montage und Fertigung im Teilbereich der *Aktivitäten / Aufgaben*. Ferner stellen Klassifikationssystemen Informationen für die einheitliche Beschreibung und Spezifikation von *Gegenständen* im Produktionsumfeld bereit.

Für die Modellierung von dynamischen Arbeitsplänen zur Ermöglichung einer individuellen Bearbeitung von Arbeitsaufträgen sind die beschriebenen Modelle nur in eingeschränktem Maße nutzbar. Zwar stellen die Klassifizierungssysteme umfängliche Informationen in strukturierter Weise bereit, jedoch fehlt genau diese umfängliche Struktur für den Bereich der *Aktivität / Aufgabe*. Da dieser Umstand bereits durch den VDI festgestellt wurde ist eine Überarbeitung der Richtlinie angekündigt, sodass mit einem geeigneten Modell in der kommenden Zeit zu rechnen ist. Für die Zielsetzung im Rahmen dieser Arbeit und die Beschränkung auf eine prototypische Umsetzung genügen die aktuell zur Verfügung stehenden Modelle und Informationen.

## 2.3 Modellierungssprachen und Verhaltensmodelle

Um die Arbeitsweise und das Verhalten von Systemen darzustellen werden Modellierungsmethoden und -sprachen verwendet. Je nach Aufgabe und Ziel der Modellierung lassen sich fachliche Modelle, zur Förderung des Systemverständnisses, und ausführbare Modelle unterscheiden. Insbesondere durch die komplexen Verhaltensmerkmale, von beispielsweise flexiblen und dezentralen Systemen, besteht bei Anwendern und Implementierern ein Interesse an Modellen, die sowohl für die fachliche als auch die ausführungsnahе Modellierung geeignet sind.

Im Verlaufe dieses Kapitels werden bekannte Modellierungssprachen und Verhaltensmodelle vorgestellt, um diese hinsichtlich ihrer Nutzbarkeit im Rahmen dieser Arbeit zu bewerten. Hierzu gehören die Ereignisgesteuerte Prozesskette (2.3.1) und die Business Process Modelling Notation (2.3.2) aus dem Bereich der Geschäftsprozesse. Im Bereich der technischen und ausführbaren Modelle werden Zustandsmaschinen (2.3.3 und 2.3.4), Petri-Netze (2.3.5) sowie die probabilistischen Hidden Markov Modelle (2.3.6) betrachtet. Den Abschluss des Kapitels bilden eine Zusammenfassung und Gegenüberstellung der präsentierten Modellierungsmethoden (2.3.7).

### 2.3.1 Ereignisgesteuerte Prozesskette (EPK)

Die Ereignisgesteuerte Prozesskette (EPK) wurde erstmals im Jahr 1992 von KELLER ET AL. vorgestellt und ist als Zusammenarbeit zwischen der SAP AG und dem Institut für Wirtschaftsinformatik der Universität des Saarlandes entwickelt worden [KNS92; NR02]. Die EPK wurde daraufhin zu einer der Hauptkomponenten der Architektur Integrierter Informationssysteme (ARIS) zur Beschreibung der Steuerungs- bzw. Prozesssicht [Sc01; Sc98] sowie zu einem Bestandteil der SAP Referenzmodelle zur Dokumentation von Geschäftsmodellen [Ke99]. Die langjährige Verfügbarkeit und das ARIS-Modellierungswerkzeug haben dazu beigetragen, dass die EPK zu den verbreitetsten Prozessmodellierungstechniken gehört [BPV12].

EPK bestehen in ihrem Grundprinzip aus Funktionen, Ereignissen sowie deren Verkettung über Kanten und Verknüpfungsoperatoren. Funktionen repräsentieren dabei eine aktive Komponente, wie eine Aktivität oder Tätigkeit innerhalb des Prozesses. Ereignisse hingegen sind passive Komponenten und stellen einen eingetretenen Zustand dar. Die Modellierung des Prozesses erfolgt durch die Verkettung von Ereignissen und Funktionen, wobei die Ausführung einer Funktion durch das Eintreten eines Ereignisses ausgeführt wird. Zur Abbildung von komplexeren Prozessen, wie parallelen Teilabläufen, werden Verknüpfungsoperatoren verwendet, die den Prozessfluss aufspalten und zusammenführen oder zu alternativen Teilabläufen führen [Be18a; BPV12]. Abbildung 2.21 zeigt die Symbole zur Modellierung einer EPK sowie den Ausschnitt einer EPK zur „Bearbeitung eines Kreditantrages“ nach [NR02]. Durch die verwendeten Verknüpfungsoperatoren werden entweder der linke Prozessablauf oder beide Prozessabläufe durchge-

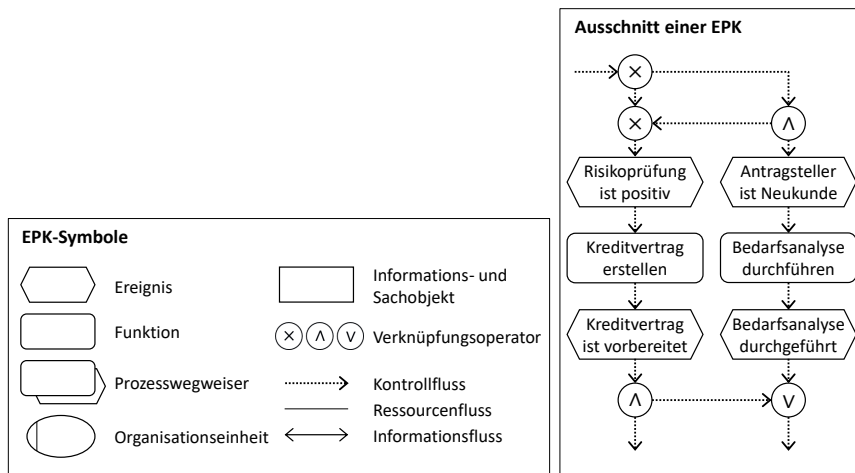


Abbildung 2.21: Ausschnitt einer EPK am Beispiel „Bearbeitung eines Kreditantrages“ (Quelle: [NR02])

führt. Der Verknüpfungsoperator  $\times$  steht dabei für eine Entweder-Oder-Operation, wohingegen  $\wedge$  den Prozessfluss aufspaltet und  $\vee$  diesen wieder zusammenführt.

Als Ergänzung der EPK existiert die erweiterte EPK (eEPK), die zusätzlich zum Prozessmodell eine Annotation mit verwendeten Ressourcen, zuständigen Organisationseinheiten und zu verwendenden Anwendungssystemen ermöglicht [BPV12]. EPK und eEPK bieten dementsprechend Elemente, um einen Geschäftsprozess auf abstrakter Ebene zu beschreiben und zu modellieren. Zur Automatisierung von Prozessen können EPK und eEPK jedoch nicht ohne weiteres verwendet werden, da Ereignisse nur verbal beschrieben sind und eine Auswertung der Verzweigungen nicht durchgeführt werden kann [Be18a], sodass diese in andere Modelle, zum Beispiel Petri-Netze oder die Business Process Modeling Notation, transformiert werden müssen.

### 2.3.2 Business Process Modeling Notation (BPMN)

Die Business Process Modeling Notation (BPMN) ist eine Prozessmodellierungssprache, die von der Business Process Management Initiative (BPMI) entwickelt und 2004 veröffentlicht wurde [BPV12]. Die aktuelle Version zum Zeitpunkt dieser Arbeit ist BPMN 2.0. Die wesentlichen Neuerungen seit der Veröffentlichung im Jahr 2004 sind die deutliche Erweiterung des Sprachumfangs sowie die Integration von ausführbaren Elementen [Ga17; SWD10]. Eine wesentliche Zielsetzung der BPMN ist die verständliche Darstellung von Prozessmodellen, die für eine technische Verwendung in automatisierten Systemen geeignet ist. BPMN ist daher einerseits für die Erstellung von rein fachlichen Modellen aber auch von ausführungsnahen Modellen geeignet und hat sich andererseits zum Standard für die Prozessmodellierung auf breiter Basis durchge-

setzt [Al15]. Trotz der gemeinsamen Notation für fachliche und für ausführungsnah Modelle unterscheiden sich BPMN-Modelle in der Praxis jedoch deutlich voneinander, da bei den fachlichen Modellen das grundlegende Verständnis der Prozesse im Vordergrund steht und auf die Einbindung zu vieler Details verzichtet wird [Al15].

Die zur Darstellung von BPMN-Modellen verwendete Basisnotation ist bereits seit langem bekannt und wurde schon in BPMN 1.0 eingeführt [Wh04]. Im Kern besteht die Basisnotation aus Ereignissen, Aktivitäten und der Verknüpfung dieser Elemente zu einem Prozess. Die Basisnotation mit ihren Elementen, der jeweiligen Beschreibung und dem entsprechenden Symbol ist in Abbildung 2.22 dargestellt. Zur Strukturierung und Klassifikation der Modelle werden in Anlehnung an die Swimlane-Methodik sogenannte Pools und Lanes verwendet [Ga17]. Ein Pool stellt dabei einen vollständigen Prozess einer Organisation dar, während Lanes diesen Prozess entsprechend verschiedener Organisationseinheiten, Rollen oder IT-Systeme weiter untergliedern (vgl. [aOMG]).


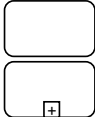




Ereignis	Ereignisse treten während eines Prozesses auf und beeinflussen diesen. Es existieren drei Standardereignisse (start, intermediate, end) sowie Spezialereignisse.	
Aktivität	Aktivitäten sind Aufgaben, die vom Unternehmen durchgeführt werden. Eine Aktivität kann atomar oder aus Subprozessen zusammengesetzt sein.	
Entscheidung	Entscheidungspunkte kontrollieren den Prozessverlauf, indem sie diesen aufspalten, zusammenführen oder entsprechend einer Entscheidung leiten.	
Kontrollfluss	Der Kontrollfluss stellt die Sequenz der Aktivitäten dar.	
Nachrichtenfluss	Der Nachrichtenfluss zeigt den Austausch von Nachrichten zwischen unabhängigen Objekten.	
Verbindung	Die Verbindung wird verwendet um Daten, Texte und andere Artefakte mit Objekten des Kontrollflusses zu verknüpfen.	

Abbildung 2.22: Basisnotation der BPMN nach [Wh04]

Abbildung 2.23 zeigt beispielhaft das BPMN-Modell des kollaborativen Prozesses einer Arztanfrage in Anlehnung an [Wh04]. Die Patientensicht und die Praxissicht zeigen verschiedene Sichtweisen auf denselben Prozess und sind daher jeweils innerhalb eines Pools dargestellt. Die Interaktion zwischen den beiden Pools erfolgt über den Nachrichtenfluss. Die einzelnen Aufgaben innerhalb eines Pools sind mittels des Kontrollflusses verknüpft. Zur weiteren Differenzierung ist der Pool *Arztpraxis* in die Lanes *Arzt* und *Empfang* unterteilt.

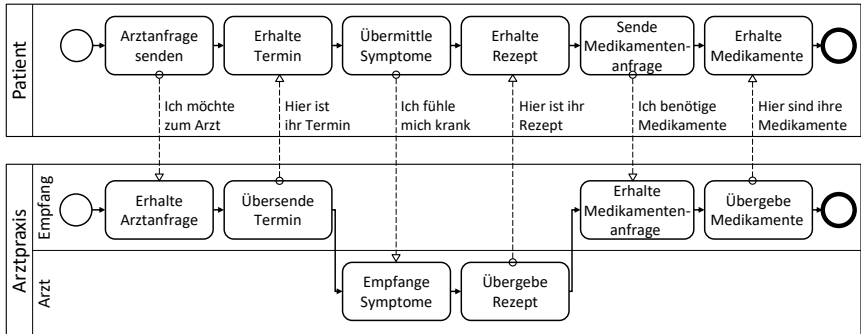


Abbildung 2.23: Beispiel für einen kollaborativen Prozess und dessen Aufteilung in Pools (in Anlehnung an [Wh04])

### 2.3.3 Endliche Zustandsautomaten

Das einfachste mathematische Modell der Automatentheorie ist der endliche Zustandsautomat (engl. finite state machine) [Si12]. Namensgebend für den Zustandsautomaten ist die endliche Anzahl an Zuständen, in denen er sich befinden kann und aus denen er nach einem festen System in einen nächsten Zustand wechselt. Die grafische Darstellung des Zustandsautomaten erfolgt als Zustandsdiagramm bestehend aus Knoten, die Zustände repräsentieren, sowie Transitionen zwischen den Knoten, die die Übergänge von einem Zustand in den nächsten darstellen (vgl. Abbildung 2.24). Für einen tiefergehenden Einblick in die Funktionsweise und den Aufbau von endlichen Zustandsautomaten empfehlen sich die Werke von HOPCROFT, SIPSER und WAGNER, auf deren Grundlagen der folgende Überblick erstellt wurde [HUM02; Si12; Wa06].

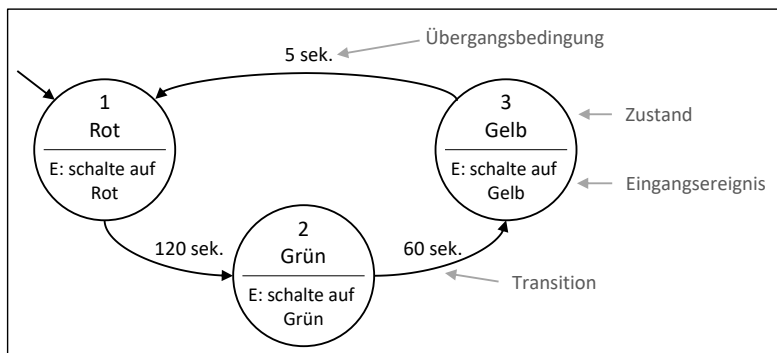


Abbildung 2.24: Endlicher Zustandsautomat am Beispiel einer Ampelsteuerung

Eine einfache Ampelsteuerung, wie in Abbildung 2.24 gezeigt, stellt ein gutes Beispiel für einen endlichen Zustandsautomaten dar. Je nach Zeitpunkt befindet sich die Ampel in einem der drei Zustände *Grün*, *Rot* oder *Gelb*. Ein Wechsel zwischen den Zuständen erfolgt in Abhängigkeit

der ausgehenden Pfeile (Transitionen) und einer entsprechenden Übergangsbedingung, die mit der Transition verknüpft ist. Für unser Beispiel heißt das, nach 120 Sekunden im Zustand *Rot* wechselt die Ampel in den Zustand *Grün*, nach weiteren 60 Sekunden in den Zustand *Gelb* und nach wiederum fünf Sekunden zurück in den Zustand *Rot*.

Während die Zustände des Systems durch die Knoten des Zustandsautomaten und die Rahmenbedingungen für den Wechsel durch die Transitionen sowie die Übergangsbedingung dargestellt werden, fehlen zur Modellierung des Systemverhaltens noch Ausgangssignale, die die tatsächliche Ampel steuern, um einen spezifischen Zustand zu erreichen. In unserem Beispiel werden diese Ausgangssignale durch ein Ereignis beim Eintritt in einen neuen Zustand erzeugt (Eingangsereignis). Das heißt, sobald das Zustandssignal in den Zustand *Gelb* wechselt, wird die tatsächliche Ampel auf *Gelb* umgeschaltet. Je nach Modellierung des Systems werden zwischen Systemen mit Eingangsereignissen  $E$  und Ausgangsereignissen  $X$  (*Moore*-Modell) oder mit Übergangsereignissen (*Mealy*-Modell) unterschieden. Im Gegensatz zu Ein- oder Ausgangsereignissen sind Übergangsereignisse völlig unabhängig von den Zuständen des Systems. Übergangsereignisse werden einer Transition zugeordnet und bei einem Zustandswechsel durch die entsprechende Transition ausgelöst.

**Definition 1** Ein endlicher Zustandsautomat wird durch das Quintupel  $(Q, \Sigma, \delta, q_0, F)$  beschrieben, wobei

$Q$  eine endliche Menge an Zuständen,

$\Sigma$  eine endliche Menge an Eingangssignalen,

$\delta : Q \times \Sigma \rightarrow Q$  eine Übergangsfunktion,

$q_0 \in Q$  der Startzustand und

$F \subseteq Q$  eine Menge an Endzuständen repräsentieren.

Die Einfachheit des mathematischen Modells ergibt sich aus der endlichen Menge an Zuständen, die nach einem einheitlichen System und mithilfe von Übergangsfunktionen und Übergangsbedingungen verkettet werden. Ferner befindet sich das Modell immer genau in einem Zustand, wodurch eine einfache Interpretation des Übergangsverhaltens möglich ist. Nebeläufige Prozesse, das heißt unabhängige und gleichzeitig ablaufende Teilprozesse, lassen sich mit diesem Modell eines Zustandsautomaten jedoch nicht repräsentieren.

### 2.3.4 UML-Zustandsmaschine

Eine weitere Modellierung von Automaten ist die UML-Zustandsmaschine, die auf Zustandsdiagrammen aus dem Bereich der Unified Modeling Language (UML) aufbaut. Einen umfassenden Einblick in die vereinheitlichte, grafische Modellierungssprache UML sowie den Aufbau von UML-Zustandsmaschinen bieten [Ha87; Ke11; Se12].



In Anlehnung an die endliche Zustandsmaschine aus Abschnitt 2.3.3 besteht die UML-Zustandsmaschine aus Knoten und aus Transitionen. Jedoch stellen Knoten der UML-Zustandsmaschine nicht nur Zustände des Systems, sondern auch sogenannte Pseudo-Zustände dar. Während Zustände weiterhin die Situation des Systems modellieren, repräsentieren Pseudo-Zustände verschiedene Steuerelemente zur Beeinflussung des Ablaufverhaltens. Zu den einfachsten Pseudo-Zuständen gehören der Start- und der Endzustand. Ein weiterer Unterschied der UML-Zustandsmaschine gegenüber der endlichen Zustandsmaschine liegt in der Möglichkeit, verschachtelte und komplexe Zustände zu modellieren, die aus mehreren Unter-Zuständen bestehen.

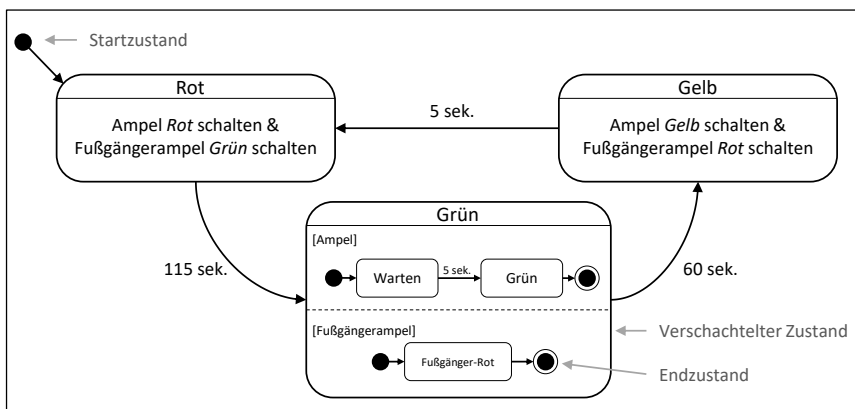


Abbildung 2.25: UML-Zustandsmaschine am Beispiel einer Ampelschaltung

Abbildung 2.25 zeigt eine UML-Zustandsmaschine mit Zuständen, Pseudo-Zuständen und einem verschachtelten Zustand in Anlehnung an das Ampelschaltungsbeispiel (vgl. Abschnitt 2.3.3). Das übergeordnete Verhalten des Automaten entspricht dabei weiterhin dem bekannten Verlauf mit dem Wechsel zwischen den Zuständen *Rot*, *Grün* und *Gelb* sowie den zeitgebundenen Übergangsbedingungen zwischen diesen Zuständen. Gegenüber dem bekannten Ampelbeispiel erfolgt jedoch zusätzlich die Steuerung einer Fußgängerampel, die im Zustand *Rot* auf Grün geschaltet und im Zustand *Gelb* auf Rot geschaltet ist. Zusätzlich ist die Grün-Phase als verschachtelter Zustand mit zwei Regionen modelliert. Eine Region entspricht dabei einer Art eigenständigen UML-Zustandsmaschine, die einen eigenen Zustand annehmen kann. Durch mehrere Regionen erreicht der verschachtelte Zustand die Modellierung von nebenläufigen, unabhängigen Prozessen. Für das vorliegende Beispiel bedeutet das, dass die Region zur Ampelsteuerung in den Zustand *warten* wechselt, während die Region zur Steuerung der Fußgängerampel zeitgleich in den Zustand *Fußgänger-Rot* wechselt. Unabhängig von der Fußgängerampel wechselt die Ampelsteuerung nach fünf Sekunden in den Zustand *Grün*. Der Vorteil von verschachtelten Zuständen liegt somit in einer Unterteilung von Zuständen in sequenzielle und nebenläufige Teilprozesse, um ein komplexes Systemverhalten modellieren zu können.

Im Gegensatz zur endlichen Zustandsmaschine in Abschnitt 2.3.3 handelt es sich bei der UML-Zustandsmaschine nicht um ein mathematisches Modell, sondern um ein Modell zur Entwicklung von Software und anderen Systemen. Die Modellierung der UML-Zustandsmaschine erfolgt somit in erster Linie grafisch.

### 2.3.5 Petri-Netze

Ein weiteres mathematisches Modell zur Beschreibung von Systemen ist das Petri-Netz, das Carl Adam Petri in den 1960er Jahren aufbauend auf endlichen Automaten entwickelte. In der heutigen Zeit werden Petri-Netze in einer Vielzahl an Disziplinen wie der Informatik [JV13], dem Maschinenbau [Ab13], der Biologie [KRS10] oder im Bereich der Geschäftsprozesse [VS11] verwendet und dienen dort vorwiegend der Beschreibung diskreter und verteilter Systeme. Ein entscheidender Unterschied von Petri-Netzen gegenüber den endlichen Automaten ist deren Prinzip zur Beschreibung von nebenläufigen Prozessen und Schaltvorgängen. Während an dieser Stelle vor allem ein Überblick über wichtige Aspekte und die Modellierung von Petri-Netzen gegeben wird, seien dem interessierten Leser die Werke von BAUMGARTEN, BRANDAU, REISIG UND STARKE für ausführlichere Einblicke in die Theorie der Petri-Netze empfohlen [Ba96a; Br19; Re10; St90].

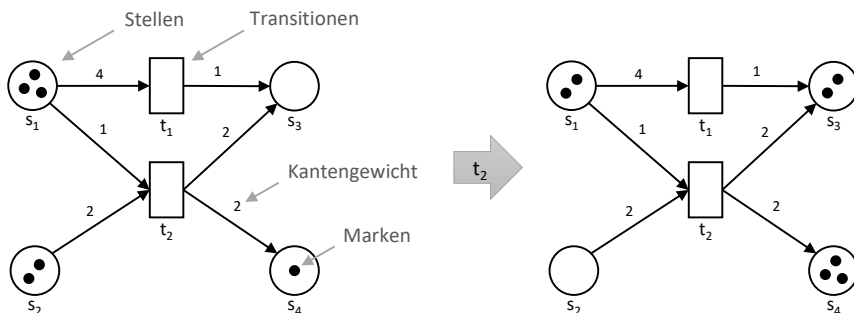


Abbildung 2.26: Petri-Netz: Schaltvorgang einer Transition

Die Darstellung von Petri-Netzen erfolgt als bipartiter Graph, das heißt als Graph, der aus zwei Arten von Knoten besteht. Bei Petri-Netzen werden diese Knoten als *Stellen* und *Transitionen* bezeichnet (vgl. Abbildung 2.26). Die Knoten des Netzes sind über Kanten miteinander verbunden, wobei jede Kante genau eine *Transition* und eine *Stelle* verbindet. Um ein System mithilfe eines Petri-Netzes zu modellieren, werden die möglichen Systemzustände durch *Stellen* dargestellt. Durch die Kopplung zweier *Stellen* über eine *Transition* werden Aktionen oder Ereignisse dargestellt, die von dem einen Zustand in den anderen führen. Ein solcher Wechsel zwischen zwei Zuständen wird auch als Schalten der verbindenden *Transition* bezeichnet. Abbildung 2.26 zeigt den Schaltvorgang der *Transition*  $t_2$  sowie die Veränderung des Petri-Netzes aufgrund dieses

Schaltvorganges. Welcher Systemzustand im Petri-Netz vorherrscht, wird mithilfe von *Marken* symbolisiert, die einer *Stelle* zugewiesen sind. Beim Schalten einer Transition wandern diese *Marken* entsprechend der verbindenden *Kanten* von einer *Stelle* in eine andere Stelle.

**Definition 2** Ein Petri-Netz wird durch das Quintupel  $(S, T, F, W, m_0)$  beschrieben, wobei

$S$  die endliche Menge aller Stellen,

$T$  die endliche Menge aller Transitionen,

$F \subseteq (S \times T) \cup (T \times S)$  die Flussrelation als Menge aller Kanten,

$W : F \rightarrow \mathbb{N}$  die Kantengewichtungsfunktion und

$m_0 : S \rightarrow \mathbb{N}_0$  die Anfangsmarkierung darstellt.

Neben der einfachen Darstellbarkeit von Systemen ist ein weiterer Vorteil von Petri-Netzen die Repräsentation in einer mathematischen Form (vgl. Definition 2), wodurch eine automatisierte Analyse der Petri-Netz-Eigenschaften und des Übergangsverhaltens zwischen Zuständen möglich ist [Ma84; Mu89; Va79].

**Definition 3** Für einen Knoten  $x \in (S \cup T)$  bildet

$\bullet x = \{y \mid (y, x) \in F\}$  den Vorbereich und

$x \bullet = \{y \mid (x, y) \in F\}$  den Nachbereich des Knotens  $x$ .

Die erweiterte Modellierung der Systemdynamik innerhalb des Petri-Netzes, das heißt wann ein Übergang zwischen Zuständen stattfindet, wird mittels *Kantengewichten* durchgeführt. Schaltet eine *Transition* und initiiert sie damit einen Wechsel, bestimmt das *Kantengewicht*, wie viele *Marken* aus einer *Stelle* entnommen werden und wie viele *Marken* der neuen *Stelle* zugefügt werden. Für das Beispiel aus Abbildung 2.26 bedeutet das, dass eine *Marke* aus der Stelle  $s_1$  und zwei *Marken* aus der Stelle  $s_2$  entnommen werden. Gleichzeitig werden jeweils zwei *Marken* der Stelle  $s_3$  und  $s_4$  hinzugefügt. Darüber hinaus bestimmt das *Kantengewicht*, wie viele *Marken* transportiert werden müssen, sodass eine *Transition* schaltfähig ist (vgl. Definition 4). Stehen, wie bei der dargestellten *Transition*  $t_1$ , nicht genug *Marken* zur Verfügung, ist die *Transition* nicht schaltfähig. Zur Analyse des Schaltverhaltens von *Transitionen* werden daher die Menge aller direkt davor liegenden Knoten (Vorbereich) und die Menge aller direkt nachfolgenden Knoten (Nachbereich) betrachtet.

**Definition 4** Eine Transition  $t$  ist genau dann schaltfähig, wenn

$$\forall s \in \bullet t : m(s) \geq W(s, t)$$

### 2.3.6 Hidden Markov Modelle

Zur Betrachtung von Prozessen, die in keiner festen deterministischen Reihenfolge ablaufen, werden Modelle aus der Stochastik verwendet. Ein bekanntes Modell zur Beschreibung solcher zeitdiskreter Zufallsprozesse sind die Markov-Ketten und das darauf aufbauende Hidden Markov Model (HMM) zur Beschreibung von verborgenen, nicht direkt beobachtbaren Zuständen. Beispiele für die Verwendung von HMM liegen unter anderem in der Aktivitätserkennung [YOI92], der Spracherkennung [JR91] oder der Biologie [Ed96]. Zur Vorstellung des HMM wird im Folgenden die Markov-Kette erster Ordnung und darauf aufbauend das HMM erläutert sowie dessen Unterschiede dargestellt. Tiefergehende Einblicke in die Markov Modelle sowie deren Anwendung bieten die Werke von KARLIN, RABINER, JUANG UND YAMATO [Ka14; Ra89; RJ86; YOI92].

Ein illustratives Beispiel zur Veranschaulichung von Markov-Ketten ist die Wettervorhersage. Dabei wird angenommen, dass für das Wetter die drei Zustände *sonnig*, *regnerisch* sowie *bewölkt* existieren und dass der aktuellen Zustand des Wetters durch ein Blick in den Himmel beobachtet werden kann. Weiterhin wird angenommen, dass aus Beobachtungen der Vergangenheit bekannt ist, mit welcher Wahrscheinlichkeit sich das Wetter zwischen diesen Zuständen ändert. Soll mithilfe dieses Wissens das Wetter vorhergesagt werden, wird das Wetter der vergangenen Tage betrachtet und auf dieser Beobachtung geschätzt, welches Wetter am nächsten Tag herrscht. In Abhängigkeit davon wie viele Tage in die Vergangenheit geschaut wird, kann diese Methode als Markov-Kette dargestellt werden. Wird das Wetter auf Basis der letzten beiden Tage geschätzt, entspricht dies einer Markov-Kette dritter Ordnung. Wird der heutige und gestrige Tag verwendet, so handelt es sich um eine Markov-Kette zweiter Ordnung.

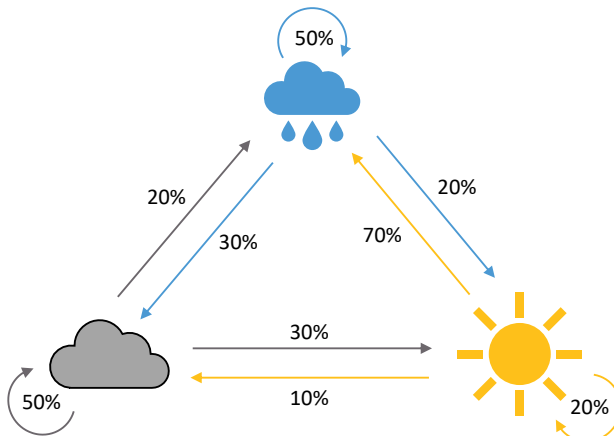


Abbildung 2.27: Markov-Prozess am Beispiel einer Wettervorhersage in Anlehnung an [Fo98]

Eine Besonderheit bietet die in Abbildung 2.27 dargestellte Markov-Kette erster Ordnung, bei der eine Vorhersage des nächsten Zustandes nur auf Basis des aktuellen Wetters erfolgt, da die-

se die Grundlage zum Aufbau von HMM sind. Die Darstellung von Markov Modellen erster Ordnung erfolgt in mathematischer Form nach Definition 5. Entsprechend des Beispiels besteht das Modell aus einer Menge an Zuständen  $S$  und einer Übergangsmatrix  $A$  zur Erfassung der Übergangswahrscheinlichkeiten zwischen den Zuständen. Für den Startzustand enthält das Modell zusätzlich die Anfangswahrscheinlichkeit  $\pi$  für jeden Zustand.

**Definition 5** Ein Markov Model erster Ordnung ist beschrieben durch das Tripel  $(S, A, \pi)$ , wobei

$S = \{s_1; \dots; s_n\}$  die Menge aller Zustände,

$A : (S \times S) \rightarrow R$  die Übergangswahrscheinlichkeit und

$\pi : S \rightarrow R$  die Anfangswahrscheinlichkeit ist.

Im Gegensatz zum beschriebenen Wetterbeispiel existieren auch Systeme, bei denen es nicht möglich ist, den Systemzustand direkt zu beobachten. Stattdessen können nur die Auswirkungen des Systemzustandes beobachtet werden, um auf den tatsächlichen Zustand zurückzuschließen. Zur Beschreibung eines solchen Systems können HMM verwendet werden.

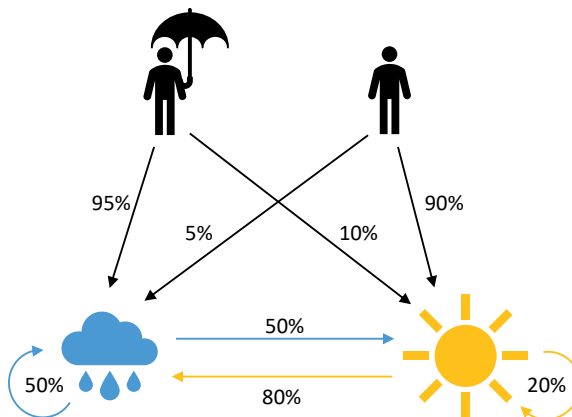


Abbildung 2.28: Beispiel eines HMM zur Bestimmung des Wetters

Zur Veranschaulichung wird das bekannte Wetterbeispiel aus Abbildung 2.27 verwendet, wobei sich auf die Zustände *sonnig* und *regnerisch* beschränkt wird. Weiterhin wird angenommen, dass ein Gefangener in einer Gefängniszelle ohne Fenster herausfinden möchte, welches Wetter draußen herrscht. Da dieser das Wetter nicht direkt beobachten kann, achtet er stattdessen auf seine Wärter. Durch die Beobachtung mehrerer Wärter findet er heraus, dass an manchen Tagen viele Wärter mit einem Regenschirm kommen. An anderen Tagen hat kaum ein Wärter einen Regenschirm dabei. Um anhand dieser Beobachtungen auf das Wetter zu schließen, überlegt er sich das in Abbildung 2.28 dargestellte Modell. Dementsprechend kommen an einem regnerischen Tag mit einer 95 % Wahrscheinlichkeit ein Großteil der Wärter mit Regenschirm und nur

mit 5 % Wahrscheinlichkeit ein Großteil der Wärter ohne Regenschirm. An sonnigen Tagen sind es zu 90 % Wahrscheinlichkeit ein Großteil der Wärter ohne Schirm und zu 10 % Wahrscheinlichkeit ein Großteil der Wärter mit Schirm.

Gegenüber Markov-Ketten erster Ordnung enthält das HMM zusätzlich eine Menge an möglichen Beobachtungen  $V$  sowie die Zuordnungsmatrix  $B$ , die angibt mit welcher Wahrscheinlichkeit eine Beobachtung  $v_i$  in einem Zustand  $s_i$  gemacht wurde.

**Definition 6** Ein HMM ist gegeben durch das Quintupel  $(S, V, A, B, \pi)$ , wobei

$S = \{s_1; \dots; s_n\}$  die Menge aller Zustände,

$V = \{v_1; \dots; v_m\}$  die Menge Beobachtungen,

$A : (S \times S) \rightarrow R$  die Übergangswahrscheinlichkeit,

$B : (S \times O) \rightarrow R$  die Beobachtungswahrscheinlichkeit und

$\pi : S \rightarrow R$  die Anfangswahrscheinlichkeit ist.

### 2.3.7 Zusammenfassung und Zwischenfazit

Im Rahmen dieser Arbeit wird die Modellierung von flexiblen Arbeitsprozessen betrachtet, mit der einerseits die kausalen Zusammenhänge des Arbeitsprozesses und andererseits der individuelle und flexible Wechsel zwischen Arbeitsschritten berücksichtigt und repräsentiert werden kann. Im Laufe dieses Kapitels wurden bekannte Modellierungsmodelle präsentiert, die hinsichtlich ihrer Nutzbarkeit für eine Modellierung flexibler Arbeitsprozesse betrachtet werden. Während eine finale Bewertung der Modelle in Kapitel 4.3.1 durchgeführt wird, werden die Modelle gegenübergestellt und mittels der folgenden fünf Kriterien eingeteilt:

1. Modellierung nebenläufiger Teilprozesse
2. Abbildung deterministischer Prozesse
3. Modellierung zufallsbasierter Prozesse
4. Betrachtung verborgener, nicht direkt beobachtbarer Zustände
5. Ausführbarkeit der Modelle

Das Ergebnis der Gegenüberstellung ist in Abbildung 2.29 dargestellt. Die Einteilung der Modelle entsprechend der jeweiligen Kriterien erfolgt dabei nach den Stufen „geeignet“, „teilweise geeignet“ und „ungeeignet“, wobei keine explizite Unterteilung zwischen den Stufen „geeignet“ und „teilweise geeignet“ mittels Entscheidungskriterium besteht. Stattdessen soll die Gegenüberstellung einen Überblick über die jeweiligen Verwendungsbereiche geben, sodass eine Unterscheidung auf Basis der primär fokussierten Modellierungsziele und Verwendungsbereiche stattfindet.

	Nebenläufigkeit	Deterministische Prozesse	Zufallsprozesse	Verborgene Zustände	Ausführbarkeit
EPK	●	●			
BPMN	●	●			●
FSM	●	●	●		●
UML-Zustandsmaschine	●	●			
Petri-Netz	●	●	●		●
HMM			●	●	●

● geeignet      ● teilweise geeignet

Abbildung 2.29: Gegenüberstellung der vorgestellten Modellierungsmethoden

## 3 Methodisches Vorgehen und Anforderungsdefinition

Das Ziel dieses Kapitels ist es, die Vorbereitungen zur Beantwortung der Forschungsfragen und zur Entwicklung eines kontextsensitiven, modularen Assistenzsystems darzulegen. Hierzu werden im ersten Schritt die gewählte Methodik zur Bearbeitung des Themas vorgestellt sowie die Phasen dieser Methodik näher erläutert (Abschnitt 3.1). Ferner erfolgt eine Abgrenzung des betrachteten Systemraumes in Abschnitt 3.2, um den Fokus dieser Arbeit festzulegen sowie im Rahmen dieser Ausarbeitung berücksichtigte und nicht berücksichtigte Aspekte zu bestimmen. Den Abschluss dieses Kapitels bildet eine Analyse der globalen und allgemeinen Anforderungen an die Entwicklung des Systems (Abschnitt 3.3).

### 3.1 Meta-Methodik für die Bearbeitung der Zielsetzung

Die übergeordnete Methodik zur Entwicklung des kontextsensitiven, modularen Assistenzsystems gliedert sich in die drei logisch aufeinanderfolgenden Phasen:

1. *Anforderungsanalyse und Informationsakquise* (Kapitel 3)
2. *Synthese und Modellbildung* (Kapitel 4)
3. *Umsetzung und Evaluation* (Kapitel 5)

Das Ziel dieser Phasenaufteilung ist die Erarbeitung von allgemeinen und globalen Anforderungen an das Gesamtvorhaben sowie von funktionalen und spezifischen Anforderungen an Teilkomponenten des zu entwickelnden Systems (1. *Anforderungsanalyse und Informationsakquise*). Aufbauend auf diesen Anforderungen und Rahmenbedingungen erfolgt eine Synthese der gesammelten Informationen und die Modellierung des kontextsensitiven, modularen Assistenzsystems als Basis für eine praktische Umsetzung (2. *Synthese und Modellbildung*). Die prototypische Umsetzung und Implementierung des allgemein modellierten Systems bildet den Abschluss und erfolgt zusammen mit einer Evaluation am Beispiel eines konkreten Anwendungsszenarios (3. *Umsetzung und Evaluation*). Die Durchführung der einzelnen Phasen orientiert sich an zwei bekannten Entwicklungsmethodiken, die in Abbildung 3.1 als kombiniertes Modell dargestellt sind.



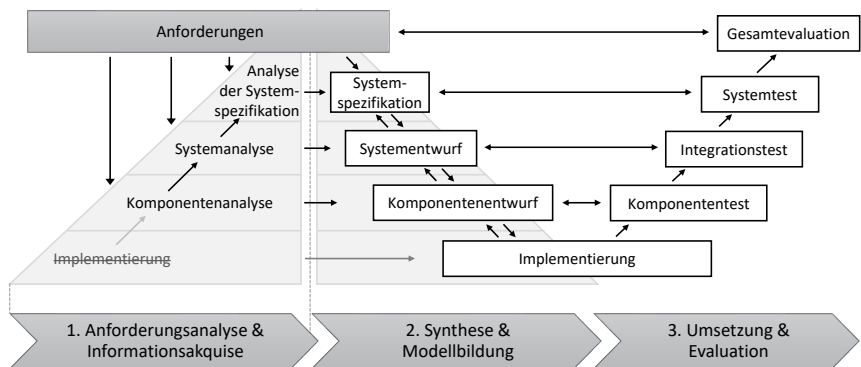


Abbildung 3.1: Meta-Methodik für die Entwicklung des kontextsensitiven, modularen Assistenzsystems

Für die Phasen der *Anforderungsanalyse und Informationsakquise* bedient sich diese Arbeit den Prinzipien des Software Reverse Engineerings (SRE) [Ba96b; Va17], um aufbauend auf bereits existierenden Assistenzsystemen relevante Anforderungen an die Entwicklung des kontextsensitiven, modularen Assistenzsystems abzuleiten. Unter der Methode des SRE wird das Zerlegen und Analysieren eines Systems oder einer Softwarekomponente in Einzelteile verstanden, um dessen Struktur und die Funktionsweise der Software zu verstehen und modellieren zu können. Die Ziele hinter der Anwendung von SRE sind beispielsweise die Rückgewinnung von verlorengegangenen Quellcode, die Rekonstruktion eines Systems oder aber die Neugestaltung und Verbesserung eines bekannten Systems im Rahmen eines Reengineerings [Ba96b; RH96]. Je nach Ziel der Anwendung unterscheiden sich die Vorgehensweise und die für den Prozess zur Verfügung stehenden Informationen. Die Gemeinsamkeit aller Vorgehensweisen ist die Durchführung in umgekehrter Entwicklungsreihenfolge, das heißt ausgehend von einer fertigen Implementierung hin zum Systemkonzept.

Im Rahmen dieser Arbeit wird das Ziel der Neugestaltung und Verbesserung eines bestehenden Systems betrachtet. Hierbei stehen alle Informationen oder Teile der Informationen hinsichtlich der ursprünglichen Anforderungen, des Systemdesigns und der Implementierung zur Verfügung. Die Bearbeitungsreihenfolge dieser Vorgehensweise startet bei der Betrachtung der fertigen Implementierung des Systems, geht über in eine Untersuchung der Komponentenentwürfe und endet mit der Betrachtung der Systemspezifikation und der Systemanforderungen. In jedem Bearbeitungsschritt wird das vorhandene Wissen analysiert und im Sinne der Neugestaltung und Verbesserung abstrahiert. Einerseits werden dadurch neue Anforderungen und Informationen generiert, die für den nächsten Bearbeitungsschritt verwendet werden. Andererseits werden diese Informationen für den äquivalenten Bearbeitungsschritt der Neuentwicklung verwendet. Die grafische Darstellung des SRE-Prozesses entspricht der zweigeteilten grauen Pyramide (in Anlehnung an [RH96]) im linken Teil von Abbildung 3.1. Die linke Hälfte der Pyramide

repräsentiert dabei den rückwärts gerichteten Prozess und die rechte Hälfte eine anschließend folgende Neuentwicklung mit vorwärts gerichtetem Prozess.

Die Entwicklung des Systems in den Phasen der *Synthese und Modellbildung* sowie der *Umsetzung und Evaluation* folgt den Prinzipien des Systems Engineerings (SE) [Hä04; Ha19]. Insbesondere baut diese Arbeit auf dem Konzept des V-Modells als Teil des SE auf. Der Übergang zwischen SRE und SE findet durch eine Überlagerung des Bereichs der vorwärts gerichteten Systementwicklung statt, die dem linken Arm des V-Modells und dem SE-Prinzip „Vom Groben zum Detail“ entspricht. Hierbei wird beginnend von den Systemanforderungen ein Grobentwurf des Systems erstellt und sich mit steigendem Detailgrad bis zur konkreten Implementierung des Systems vorwärts gearbeitet. Der rechte Arm des V-Modells repräsentiert die Validierung und Verifikation des Systems und der einzelnen Entwicklungsschritte. Hierbei darf das V-Modell jedoch nicht als reines Prozessmodell mit zeitlicher Abfolge der Teilschritte verstanden werden, sondern ebenso als Entity-Relationship-Modell, das einen logischen Zusammenhang der Teilschritte herstellt. Notwendige Anpassungen aufgrund der Systemanforderungen, die sich in späteren Teilschritten zeigen, führen zu einer Bearbeitung im entsprechend vorgelagerten Bearbeitungsschritt.

Tabelle 3.1: Phasen der Entwicklungsmethodik mit Ergebnis und Zuordnung zur Kapitelstruktur

Phase	Ergebnis der Phase	Methoden & Werkzeuge	Kapitel
1 <b>Anforderungsanalyse &amp; Informationsakquise</b>	Globale Anforderungen	Literaturrecherche Flussdiagramme	3
	Abgrenzung des Systemraums		
	Lokale Anforderungen		
	Abstrahiertes Modell aktuelle Assistenzsysteme		
2 <b>Synthese &amp; Modellbildung</b>	Modell der Systemarchitektur	Flussdiagramme	4
	Modell der Verarbeitungskomponente	UML-Diagramme	
		Separation of Concerns	
3 <b>Umsetzung &amp; Evaluation</b>	Prototypische Umsetzung des Systems	Automatisierte Tests	5
	Bewertung des entwickelten Systems	Anwendertests	

Der strukturelle Aufbau dieser Arbeiten orientiert sich an den vier definierten Phasen der Methodik. Jede Phase wird dabei in einem eigenen Hauptkapitel bearbeitet und liefert ein Teilergebnis für die weitere Entwicklung des Systems. Tabelle 3.1 bietet einen Überblick über die Zuordnung der Phasen zu den Kapiteln dieser Arbeit. Ferner werden die Teilergebnisse der einzelnen Phasen sowie die in den Phasen verwendeten Methoden und Werkzeuge aufgezeigt. Eine tiefergehende Beschreibung des methodischen Vorgehens innerhalb der jeweiligen Phasen findet sich im entsprechenden Kapitel.

## 3.2 Abgrenzung des betrachteten Problemraums

Um einerseits eine zielgerichtete Bearbeitung der Forschungsfragen und Entwicklung eines kontextsensitiven Assistenzsystems durchzuführen und andererseits das bearbeitete Forschungsthema im Kontext anderer Forschungsarbeiten einzuordnen werden eine Abgrenzung des betrachteten Problemraumes und eine Festlegung der Systemgrenzen durchgeführt. Hierzu erfolgt im ersten Schritt eine Fokussierung der Anwendungsdomäne für das Assistenzsystem, wobei der betrachtete Aufgabenbereich des Systems und die Art des Systems spezifiziert werden. Im zweiten Schritt werden die Teilsystembereiche von Assistenzsystemen aufgezeigt und der zu untersuchende Teilbereich sowie die relevanten Eingangsgrößen abgegrenzt.

### Betrachtete Anwendungsdomäne

In Anlehnung an die in Abschnitt 2.1 durchgeführte Literaturrecherche zu Assistenzsystemen sowie die Abgrenzung dieser Arbeit gegenüber verwandter Arbeiten, werden lediglich kontextsensitive Assistenzsysteme im Rahmen des Produktionsumfeldes betrachtet, die eine kognitive Unterstützung bei der Durchführung von manuellen Tätigkeiten bieten. Physische oder sensorische Unterstützungssysteme sowie Systeme, die die Aus- oder Weiterbildung (vgl. Abschnitt 2.1.1) adressieren, werden nicht näher betrachtet. Ferner erfolgt eine Fokussierung auf den Anwendungsbereich der Reparatur von technischen Systemen, das heißt dem Demontieren von Bauteilen, dem Durchführen von Reparaturoperationen und dem anschließenden Remontieren der Komponenten. Trotz der thematischen Nähe des Aufgabenbereiches zur manuellen Montage herrschen unterschiedliche Anforderungen an diese beiden Bereiche. Während im Bereich der manuellen Montage ein weitestgehend strukturiertes und vorausgeplantes Bearbeiten möglich ist, ist die Reparatur von technischen Systemen stärker durch ein intuitionsgetriebenes und individuelles Bearbeiten geprägt. Da für die manuelle Montage bereits umfängliche Arbeiten existieren (vgl. Literaturrecherche in Abschnitt 2.1) und diese weniger von flexiblen, individuellen Bearbeitungsweisen bestimmt ist, wird dieser Anwendungsbereich nicht näher fokussiert. Bekannte Ansätze und technische Methoden aus diesem Bereich fließen jedoch durch die Literaturrecherche zu Assistenzsystemen für manuelle Tätigkeiten in die Ausarbeitung mit ein.

### Betrachteter Systembereich

Werden computerbasierte Systeme betrachtet, lässt sich deren Aufbau in die Bereiche Eingabe, Verarbeitung und Ausgabe unterteilen (vgl. Abschnitt 4.2). Bezogen auf kontextsensitive Assistenzsysteme entspricht dies der folgenden Einteilung:

- *Eingabe*: Erfassung des Arbeitskontextes sowie der Aktivität über Sensorsysteme<sup>5</sup>
- *Verarbeitung*: Interpretation der Arbeitssituation und Bereitstellung von Unterstützung<sup>6</sup>
- *Ausgabe*: Nutzerzentrierte Visualisierung und Unterstützung für die nutzende Person<sup>7</sup>

<sup>5</sup>vgl. „nicht invasive Sensorsysteme“ in [AGU14] (Abschnitt 2.1.2)

<sup>6</sup>vgl. Plant@Hand in Abschnitt 2.1.2 oder Kontextsensitivität in Abschnitt 2.1.3

<sup>7</sup>vgl. motionEAP in Abschnitt 2.1.2

Während es in allen drei Systembereichen entsprechende Forschungsarbeiten mit dem Ziel der Entwicklung von praxistauglichen Assistenzsystemen gibt, fokussiert sich diese Arbeit explizit auf den Bereich der *Verarbeitung*. Da dieser jedoch direkt mit den Bereichen der *Eingabe* und *Ausgabe* verknüpft ist, erfolgt eine Betrachtung der Schnittstellen und Interaktionen zwischen *Eingabe*, *Verarbeitung* und *Ausgabe* auf Basis bestehender Assistenzsysteme (vgl. Abschnitt 4.2). Durch die Analyse der relevanten Interaktionen wird eine entkoppelte Behandlung des Bereiches der *Verarbeitung* ermöglicht (vgl. Abbildung 3.2). Ferner erfolgt eine Bearbeitung der Forschungsfragen und Berücksichtigung der folgenden Annahmen und Eingangsgrößen:

1. Die Unterstützung erfolgt für einen Arbeitsprozess bestehend aus
  - (a) Arbeitsschritten mit zugehöriger Aufgabenbeschreibung
  - (b) kausalen, konstruktionsbedingten Abhängigkeiten zwischen Arbeitsschritten
2. Die aktuelle Arbeitssituation lässt sich anhand von Kontextinformationen beschreiben (vgl. Abschnitt 2.2)

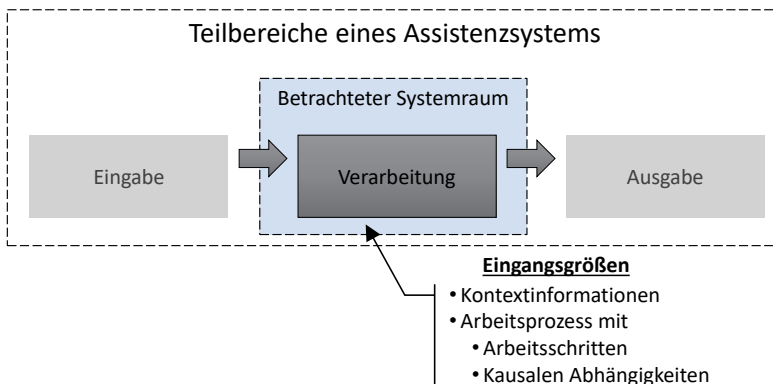


Abbildung 3.2: Teilbereiche eines Assistenzsystems und Abgrenzung des betrachteten Systemraums

### 3.3 Definition der globale Anforderungen

Entsprechend der Prinzipien des SE erfolgt eine Anforderungserhebung als wichtige Teilaufgabe, um ein umfassendes Verständnis über das Gesamtsystem zu erlangen [Hä04]. Dabei erfolgt häufig eine Unterteilung in globale und lokale Anforderungen, wobei die globalen Anforderungen eine allgemeingültige Sichtweise auf das Gesamtsystem bieten [Sc10]. Lokale Anforderungen hingegen beziehen sich auf dedizierte Teilkomponenten oder -bereiche des Systems und beschreiben die grundlegenden Anforderungen, die deren Funktionsweisen an das System stellen [RR12].

Im Rahmen dieser Arbeit orientiert sich die Erfassung der globalen Anforderungen des Systems an den Designprinzipien für die Entwicklung interaktiver Systeme [GC96] sowie der Norm für Softwarequalität [ISO IEC 25010]. Die Dokumentation erfolgt im weiteren Verlauf dieses Kapitels unter der Angabe einer ID (**GA...**), wobei die Anforderungen aus darstellungstechnischen Gründen zu thematischen Gruppen zusammengefasst wurden. Die Definition der lokalen Anforderungen erfolgt im Verlauf der Analyse bestehender Assistenzsysteme (vgl. Abschnitt 3.4). Zur Abgrenzung zu den globalen Anforderungen werden diese mit einer eigenständigen ID (**LA...**) versehen.

### **Kompatibilität und Übertragbarkeit**

Die kontextsensitive Verarbeitungskomponente stellt einen zentralen Bestandteil von Assistenzsystemen im produktiven Umfeld dar. Um eine breite Nutzbarkeit in bestehenden und bekannten Assistenzsystemen zu ermöglichen, ist es unerlässlich, dass die Verarbeitungskomponente mit diesen Systemen interagieren kann. Ferner muss das System offen und übertragbar gestaltet sein, um eine Nutzung der Prinzipien und Arbeitsweise in anderen Anwendungsbereichen oder Systemen zu ermöglichen.

#### **GA1: Integrierbarkeit in bestehende Systeme**

Die Verarbeitungskomponente soll so gestaltet sein, dass sie in bestehende und bekannte Assistenzsysteme der manuellen Montage integrierbar ist (vgl. Analyse und Abstrahierung bestehender Assistenzsysteme in Abschnitt 3.4).

#### **GA2: Übertragbarkeit und Wiederverwendbarkeit**

Die Verarbeitungskomponente und deren Bestandteile sollen so gestaltet sein, dass diese für andere Anwendungsbereiche und Anwendungsszenarien wiederverwendet oder übertragen werden können (vgl. auch **GA3**).

### **Modifizierbarkeit und Anpassbarkeit**

Das Ziel dieser Arbeit ist die Entwicklung eines modular gestalteten Assistenzsystems für die flexible Unterstützung bei Reparaturaufgaben. Sowohl die modulare Gestaltung als auch das Anwendungsfeld der Reparatur sind geprägt von veränderlichen Anforderungen an das System. So bedingt die Modularität des Systems eine Modifizierbarkeit des Systemaufbaus, bei dem unabhängige Teilkomponenten kombiniert werden können. Das Anwendungsfeld der Reparaturaufgabe ist geprägt von flexiblen Prozessen und Bearbeitungswegen, die von Produkt zu Produkt und von Situation zu Situation unterschiedlich sein können. Das auf diesen Prozessen beruhende Systemverhalten muss daher einfach veränderbar und an die Situation anpassbar sein.

#### **GA3: Modifizierbarer Systemaufbau**

Die Struktur des Systems muss aus unabhängigen Teilkomponenten bestehen, um den Systemaufbau an die gegebene Arbeitssituation anpassen und Teilkomponenten entsprechend veränderter Anforderungen modifizieren zu können (vgl. Prinzip des *Separation of Concerns* in Kapitel 4).

**GA4: Anpassbares Systemverhalten**

Das Systemverhalten wird maßgeblich durch die Prozesse beeinflusst, für die eine Unterstützung geboten werden soll. Damit eine Unterstützung für veränderte oder andere Prozesse möglich ist, muss das Systemverhalten einfach anpassbar sein.

**Gebrauchstauglichkeit und Benutzerfreundlichkeit**

Die Aufgabe von Assistenzsystemen ist die Unterstützung der nutzenden Person, sodass diese sich auf ihre Tätigkeit fokussieren kann. Darüber hinaus erfolgt der Einsatz von Assistenzsystemen in Bereichen, die von komplexen oder einer Vielzahl von Arbeitsprozessen geprägt sind. Damit Assistenzsysteme diese Aufgabe effizient erfüllen können, müssen diese über intuitive Benutzerschnittstellen verfügen, sodass die nutzende Person nicht durch die Bedienung des Systems abgelenkt wird. Zudem müssen die Interaktionen hinsichtlich ihrer Notwendigkeit und Gebrauchstauglichkeit untersucht werden, um eine effiziente Bedienung des Assistenzsystem zu ermöglichen.

**GA5: Intuitive Bedienbarkeit**

Die Benutzerschnittstellen des Assistenzsystems müssen so gestaltet sein, dass eine Interaktion zwischen der nutzenden Person und dem System intuitiv ist und von der nutzenden Person als natürliche Verhaltensweise wahrgenommen wird.

**GA6: Gebrauchstauglichkeit der Bedienoptionen**

Direkte Interaktionen mit dem Assistenzsystem müssen auf ein notwendiges Minimum reduziert werden, sodass unnötige Ablenkungen vermieden werden. Die Umsetzung der notwendigen Interaktionen müssen darüber hinaus gebrauchstauglich sein, um einen effizienten Umgang mit dem System zu erreichen.

**System- und Plattformunabhängigkeit**

In Anlehnung an die Anforderungen der Modifizierbarkeit und Anpassbarkeit erfordert ein modulares und veränderliches Assistenzsystem eine hersteller- oder plattformunabhängige Umsetzung. Durch diese kann das System in breiten Bereichen des Produktionsumfeldes eingesetzt werden. Die Unabhängigkeit der Umsetzung bezieht sich hierbei einerseits auf die direkten Implementierungsmöglichkeiten der Verarbeitungskomponente und andererseits auf die Integration mit Systemen anderer Hersteller.

**GA7: Plattformunabhängige Systemgestaltung**

Die Ausgestaltung des kontextsensitiven, modularen Assistenzsystems muss hersteller- und plattformunabhängig erfolgen, um eine individuelle Umsetzung des Systems für individuelle Anwendungssituationen zu ermöglichen (vgl. Modellbasierter Systementwurf in Kapitel 4).

**GA8: Plattformunabhängige Integration**

Um eine Integration des kontextsensitiven, modularen Assistenzsystems in Systeme verschiedener Hersteller zu ermöglichen, muss eine plattformunabhängige Interaktion zwischen Verarbeitungskomponente und dem restlichen Assistenzsystem umgesetzt werden.

**Funktionalität**

Zusätzlich zu den bereits genannten nicht-funktionalen Anforderungen stellt die Zielsetzung dieser Arbeit auch funktionale Anforderungen an die globale Funktionsweise des Assistenzsystems und damit auch an die Funktionsweise der Verarbeitungslogik. Hierzu gehört die Anpassungsfähigkeit des Assistenzsystems, um die nutzenden Personen bei einer selbstbestimmten Arbeitsweise zu unterstützen und somit eigenverantwortliches Arbeiten zu fördern. Dadurch kann die Bearbeitungsreihenfolge durch jede nutzende Person individuell gewählt werden. Gleichzeitig agiert das System als Unterstützung bei der Bearbeitung von Arbeitsprozessen mit kausal bedingten Abhängigkeiten, die nicht in jeder beliebigen Bearbeitungsreihenfolge durchgeführt werden können. Das System muss diese kausalen Abhängigkeiten erfassen und beachten, um nicht korrekte Arbeitsweisen sondern auch falsche Bearbeitungsreihenfolgen erkennen und diese aufzeigen zu können.

**GA9: Selbstbestimmung der Nutzer**

Das System muss eine Selbstbestimmung der nutzenden Personen, hinsichtlich einer freien Wahl der Bearbeitungsreihenfolge und Bearbeitungsweise, ermöglichen.

**GA10: Wahrung kausaler Abhängigkeiten**

Das System muss kausale oder konstruktionsbedingte Abhängigkeiten eines Arbeitsprozesses erfassen und verarbeiten können, um alle praktisch durchführbaren Bearbeitungswege ermitteln zu können.

### 3.4 Informationsakquise lokaler, funktionaler Anforderungen

Ziel der Informationsakquise und -aufbereitung ist es, ein abstrahiertes Modell bestehender Assistenzsysteme zu erstellen und daraus Anforderungen abzuleiten, die an die Entwicklung eines kontextsensitiven, modularen Assistenzsystems gestellt werden. Ein zentraler Aspekt ist die Berücksichtigung eines allgemeingültigen und systemkompatiblen Ansatzes (vgl. **GA1** und **GA2**), um ein modulares, austauschbares Assistenzsystem zu entwickeln und keine spezifische Einzellösung für ein System. Aus diesem Grund werden bestehende Systeme Schritt für Schritt mittels der Methode des SRE analysiert und zu einem abstrahierten Modell zusammengeführt. Der Gesamtprozess besteht dabei aus drei Analyseschritten, in denen das System mit unterschiedlichem Fokus und Detailgrad untersucht wird. Als initiale Basis dient das Assistenzsystem des Handarbeitsplatzes der *SmartFactory*<sup>KL</sup> (vgl. Vorstellung in Abschnitt 2.1.2). Der Aufbau des Systems sowie Modelle und Entwürfe des Systems stehen einerseits durch Veröffentlichungen und

andererseits durch eigene Erfahrungen während mehrjähriger Arbeit mit dem System zur Verfügung. Um ein möglichst allgemeingültiges Modell zu erhalten, erfolgt in jedem Analyseschritte ein Vergleich der Konzepte und Entwürfe des Handarbeitsplatzes der *SmartFactory*<sup>KL</sup> mit bekannten Assistenzsystemen.

Der Vergleich der Systeme erfolgt mithilfe von selbstentwickelten Erfassungsbögen, in denen die verschiedenen Systeme und ihre Besonderheiten erfasst und repräsentiert werden (vgl. Abschnitt 2.1.2). Die Erfassungsbögen sind dabei einheitlich nach einer Vorlage aufgebaut, die sich in Anlehnung an das EVA-Prinzip<sup>8</sup> und die Struktur des *SmartFactory*<sup>KL</sup>-Systems in die zentralen Kategorien *Erfassung*, *Visualisierung*, *Prozessverarbeitung* und *Prozessmodell* untergliedert. Für jede dieser Kategorien werden Informationen über das repräsentiere System gesammelt und stichpunktartig erfasst. Besonderheiten, die sich als wichtig für das Systemverhalten und den Funktionsraum der Systeme erweisen, werden in einer zusätzlichen Kategorie erfasst. Die Auswertung der Bögen, um eine abstrahierte Modellbildung zu erreichen, geschieht durch einen direkten Vergleich mit den Modellen des Handarbeitsplatzes des *SmartFactory*<sup>KL</sup>-Systems. Nicht repräsentierte oder abweichende Aspekte werden erfasst und in ein abstrahiertes Modell aufgenommen. Der vollumfängliche Bestand der Erfassungsbögen ist in Anhang A zu finden.

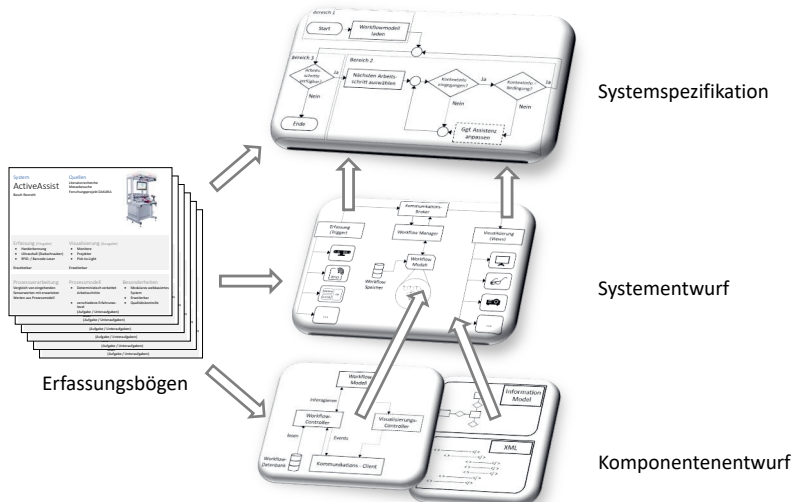


Abbildung 3.3: Vorgehensweise zur Datenakquisition und Datenauswertung auf Basis von SRE Methoden

Die Umsetzung des beschriebenen Vorgehens nach den Prinzipien des SRE ist in Abbildung 3.3 dargestellt. Der Vergleich zwischen *SmartFactory*<sup>KL</sup>-Handarbeitsplatz und weiteren Assistenzsystemen, aufbauend auf den Erfassungsbögen, wird in drei Schritten durchgeführt. In jedem

<sup>8</sup>EVA steht für Eingabe, Verarbeitung und Ausgabe. Vgl. [Dw18]



Schritt werden die Systeme bzw. die Verarbeitungsebenen der Systeme auf einem anderen Abstraktionslevel betrachtet. Zunächst erfolgt die Betrachtung auf der Ebene des Komponentenentwurfes (Abschnitt 3.4.1). Aufgrund der Zielstellung und Systemabgrenzung dieser Arbeit wird sich hierbei auf den Bereich der Verarbeitungskomponente beschränkt (vgl. Abschnitt 3.2). Ziel dieses Schritts ist die Herleitung eines verallgemeinerten Komponentenentwurfes. Weiterhin werden Anforderungen mit Bezug zur Verarbeitungslogik erfasst. Im zweiten Schritt wird ein abstrahierter Systementwurf des Gesamtsystems entwickelt (Abschnitt 3.4.2). Hieraus folgt die Definition von Anforderungen an den Systemaufbau, die Bestandteile und die Interaktionen zwischen diesen Bestandteilen. Im dritten und letzten Schritt werden die Abstrahierung und Modellierung des Gesamtsystemverhaltens (Abschnitt 3.4.3) sowie die Ableitung von Anforderungen bezogen auf dieses Systemverhalten durchgeführt. Den Abschluss der Informationsakquise und -aufbereitung bildet eine Zusammenfassung der Ergebnisse des SRE-Prozesses und eine Übersicht der gesammelten lokalen und funktionalen Anforderungen (Abschnitt 3.4.4).

### 3.4.1 Abstrahierter Komponentenentwurf

Im ersten Analyseschritt des SRE-Prozesses wird ein abstrahierter Komponentenentwurf als Meta-Modell der Verarbeitungskomponenten bestehender Assistenzsysteme hergeleitet und dargestellt. Hierzu erfolgt zuerst eine Untersuchung der Verarbeitungskomponente des Basis-systems (Handarbeitsplatz der *SmartFactory*<sup>KL</sup>, vgl. Abschnitt 2.1.2) auf Basis eigener Erfahrungen des Autors. Als Ergebnis werden die Bestandteile und die Funktionsweise der Verarbeitungskomponente dargelegt bevor ein Vergleich mit den erfassten Systemen (vgl. Erfassungsbögen in Anhang A) durchgeführt wird.

Die Untersuchung des *SmartFactory*<sup>KL</sup>-Systems zeigt, dass die Verarbeitungskomponente (Workflow-Manager genannt) primär aus zwei Elementen aufgebaut ist. Das **erste Element** repräsentiert ein Arbeitsprozessmodell bestehend aus einzelnen Arbeitsschritten, die notwendig sind, um einen Arbeitsauftrag zu bearbeiten. Der Aufbau des Modells folgt einer sequenziellen Verkettung dieser Arbeitsschritte über individuelle Ereignisse. Ein Ereignis verknüpft dabei genau zwei Arbeitsschritte unidirektional und repräsentiert den Abschluss des einen sowie den Start des anderen Arbeitsschritts (vgl. Abbildung 3.4). Die sequenzielle Abfolge der Arbeits-

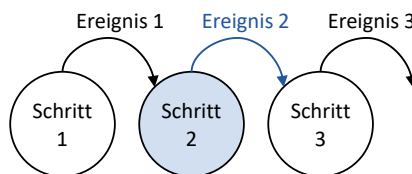


Abbildung 3.4: Schematische Darstellung des Arbeitsprozessmodells der Verarbeitungskomponente des Handarbeitsplatzes der *SmartFactory*<sup>KL</sup>

schritte ermöglicht es dem System als Zustandsmaschine zu agieren (**zweites Element**), wobei jeder Arbeitsschritt einem Zustand und jedes Ereignis einer Übergangsbedingung entsprechen (vgl. Abschnitt 2.3.3). Das heißt, bei jedem Auftreten eines neuen Ereignisses vergleicht die Verarbeitungskomponente das eingegangene Ereignis mit dem Ereignis, dass den aktuellen Arbeitsschritt mit dem nachfolgenden Arbeitsschritt verknüpft. Stimmen die Ereignisse überein, schaltet die Verarbeitungseinheit in den nächsten Arbeitsschritt weiter. Ferner wird das Gesamtsystem darüber informiert, in welchen Arbeitsschritt das System gewechselt hat.

Entsprechend der beschriebenen Vorgehensweise erfolgt die Abstrahierung und Modellierung des Komponentenentwurfs auf Basis dieser Verarbeitungskomponente und des Vergleichs mit den Systemen der Erfassungsbögen. Abbildung 3.5 zeigt das Ergebnis dieses Vorgehens als interne Sicht auf die Verarbeitungskomponente des Assistenzsystems und dessen Bestandteile. Entitäten, die mit durchgehendem Rahmen dargestellt sind, repräsentieren die beschriebenen Elemente des *SmartFactory*<sup>KL</sup>-Systems, wobei die Zustandsmaschine durch den **Workflow-Controller** und das Arbeitsprozessmodell durch das **Workflow-Modell** repräsentiert werden. Die Kommunikation mit dem Gesamtsystem erfolgt durch einen **Kommunikations-Client**. Die gestrichelten Elemente zeigen Erweiterungen und Anpassungen, die als Folge des Abstrahierungsprozesses durchgeführt wurden. Bestandteile wie eine Sensorik oder Nutzeroberfläche, durch die das Eintreten eines Ereignisses erfasst werden, sind nicht Bestandteil der Verarbeitungskomponente und werden durch den Kommunikations-Client vom Gesamtsystem bereitgestellt. Der Wechsel eines Arbeitsschritts durch die Zustandsmaschine (Workflow-Controller) wird ebenfalls über den Kommunikations-Client an das Gesamtsystem weitergemeldet.

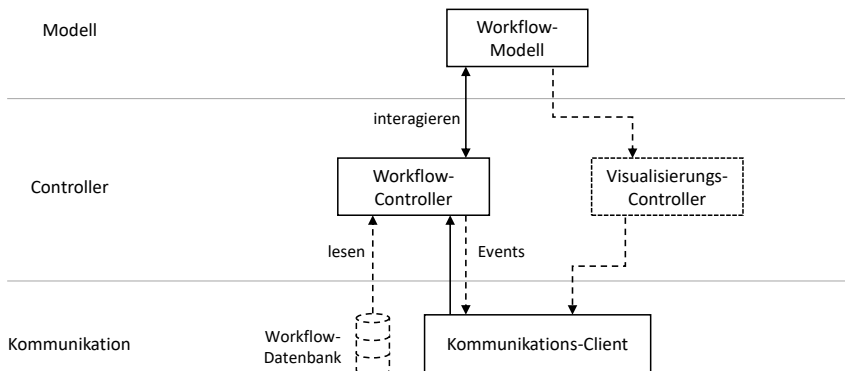


Abbildung 3.5: Abstrahierter Komponentenentwurf der Verarbeitungseinheit bestehender Assistenzsysteme

Bei den durch die Abstraktion ergänzten Elementen in Abbildung 3.5 handelt es sich um einen Visualisierungs-Controller und eine Workflow-Datenbank. Zusätzlich zeigt die Abbildung die dadurch hervorgerufenen Anpassungen an den Interaktionen zwischen diesen Bestandteilen. Eine Beschreibung der Bestandteile sowie der Anpassungen bietet die folgende Übersicht:

**Visualisierungs-Controller:** In Abgrenzung zur Verarbeitungskomponente des *SmartFactory*<sup>KL</sup>-Systems existieren Assistenzsysteme, die einen starken Fokus auf die Visualisierung von Informationen während eines Arbeitsschritts legen. Das heißt, die Unterstützungsleistung des Systems ist nicht nur vom aktuellen Arbeitsschritt, sondern auch von Situationen und Ereignissen innerhalb eines Arbeitsschritts abhängig. Die Entität des Visualisierungs-Controllers übernimmt die Aufgabe die geeignete Unterstützungsleistung zu bestimmen. Dies geschieht auf Basis des Arbeitsschritts und den Informationen zur Arbeitssituation, die über den Kommunikations-Client bereitgestellt werden. Ferner übernimmt die Komponente die Steuerung der Visualisierungselemente über den Kommunikations-Client. Beispielsysteme, aus denen diese Einheit hergeleitet wurde, sind das ProMiMo-System<sup>9</sup> und das TNO Operator Support System (vgl. [TNO; Bo17]). Beide Systeme erkennen Fehler, zum Beispiel die Entnahme aus einer falschen Materialbox, und zeigen dies der nutzenden Person an. Ein weiteres Beispiel ist das Plant@Hand-System<sup>10</sup>, dass den Detailgrad der Unterstützungsinformationen an die nutzende Person anpasst, um diese nicht mit der Bereitstellung zu vieler Informationen zu überfordern [AM16].

**Workflow-Datenbank:** Einerseits repräsentiert die Datenbankbindung den Zugriff auf verfügbare Arbeitsprozessmodelle, sodass entsprechend des Arbeitsauftrages das zugehörige Prozess-Modell geladen werden kann. Andererseits steht diese für den Zugriff auf alle Arbeitsprozessmodelle während der Laufzeit, um ähnliche Arbeitssituationen zu identifizieren und damit verknüpfte Unterstützungsfunktionen bereitzustellen. Ein Beispiel hierfür ist wiederum das Plant@Hand-System, dass durch den Zugriff auf ähnliche Arbeitsprozesse manuell hinzugefügte Zusatzinformationen dieser Arbeitsprozesse abrufen und der nutzenden Person als zusätzliche Unterstützung bereitstellen kann. Umgekehrt können eigene Zusatzinformationen erstellt und mit dem aktuellen Arbeitsschritt verknüpft werden. Diese Informationen können wiederum bei der Durchführung anderen Arbeitsprozessen abgerufen und angezeigt werden [AAU15].

Zusätzlich zu den beschriebenen Erweiterungen ergibt sich eine Anpassung bei der Bereitstellung eines Arbeitsschrittwechsels durch die Entität des Workflow-Controllers, da im verallgemeinerten Komponentenentwurf eine Erweiterung um den Visualisierungs-Controller erfolgt ist. Dieser informiert das System ebenfalls bezüglich des aktuellen Arbeitsschritts und der bereitzustellenden Unterstützungsleistungen, sodass keine fest definierte Aufgabenzuweisung hin-

<sup>9</sup>vgl. Seite 145 in Anhang A sowie [OWL; Rü14]

<sup>10</sup>vgl. Seite 144 in Anhang A

sichtlich der Informationsbereitstellung über einen Arbeitsschrittwechsel oder den aktuellen Arbeitsschritt möglich ist. Abhängig von der tatsächlichen Realisierung der Verarbeitungskomponente kann diese Aufgabe durch den Workflow-Controller oder den Visualisierungs-Controller übernommen werden.

Die funktionalen Anforderungen, die sich durch den abstrahierten Komponentenentwurf an die Entwicklung der Verarbeitungskomponente des kontextsensitiven, modularen Assistenzsystem ergeben, werden durch die Schnittstellen und die Interaktionen mit dem Gesamtsystem bestimmt. Aus diesem Grund erfolgt die Herleitung der lokalen, funktionalen Anforderungen durch die Beantwortung der folgenden Fragen:

- In welcher Art und Weise werden Interaktionen und die Kommunikation durchgeführt?
- Welche Informationen beinhalten die Interaktionen und die Kommunikation?
- Wie wird die Verarbeitung der Informationen beeinflusst?

Die Beantwortung dieser Fragen ergibt die folgenden Anforderungen an die Verarbeitungskomponente, die bei der Entwicklung der Verarbeitungslogik beachtet werden müssen.

LA1: Die Kommunikation findet eventbasiert statt

LA2: Die Eingangssignale repräsentieren Ereignisse, die durch die Sensorik erfasst werden

LA3: Die Ausgangssignale repräsentieren Informationen über den Wechsel eines Arbeitsschrittes, die ID des neuen Arbeitsschrittes und spezifische Informationen für die Visualisierung

LA4: Der Arbeitsprozess und die Übergangsbedingungen zwischen Arbeitsschritten werden in einem Workflow-Modell repräsentiert

LA5: Das zu verwendende Workflow-Modell wird durch den Workflow-Manager aus einer Datenbank geladen

### 3.4.2 Abstrahierter Systementwurf

Im zweiten Analyseschritt des SRE-Prozesses erfolgt die Herleitung eines abstrahierten Systementwurfes für bekannte Assistenzsysteme, um die lokalen und funktionalen Anforderungen für die Entwicklung hinsichtlich des Systemaufbaus und der Systemkomponenten abzuleiten. Das Vorgehen zur Abstrahierung folgt weiterhin der in Abschnitt 3.4 vorgestellten Vorgehensweise. Die Grundlage des abstrahierten Systementwurfes ist somit die Systemarchitektur des *SmartFactory*<sup>KL</sup>-Handarbeitsplatzes auf Basis der Veröffentlichung von QUINT ET AL. (vgl. Abbildung 3.6) [Qu16]. Das Gesamtsystem lässt sich hierbei in Anlehnung an das EVA-Prinzip [Dw18] oder die Arbeiten von WÖLFLE in die drei funktionalen Bereichen *Eingabe*, *Verarbeitung* und

*Ausgabe* unterteilen. Abbildung 3.6 zeigt im linken Teil der Darstellung den Bereich der *Eingabe* des SmartFactory<sup>KL</sup>-Systems, der die Erfassung von Eingangssignalen in das System abdeckt. Eine Eingabe im klassischen Sinne bei der ein Nutzer Informationen eingibt erfolgt jedoch nicht. Stattdessen wird die Eingabe passiv durchgeführt, das heißt Ereignisse und Situationen, die sich während des Arbeitsprozesses ereignen, werden durch die Sensoren im Bereich der *Eingabe* erfasst und an das System übermittelt. Die Workflow-Erkennung repräsentiert den Bereich der *Verarbeitung* des EVA-Prinzips. In diesem Bereich werden die erfassten Eingangssignale verarbeitet und die Ergebnisse an den Bereich der *Ausgabe* bereitgestellt. Die Aufgaben der Verarbeitung entsprechen dabei den in Abschnitt 3.4.1 vorgestellten Funktionsweisen und Interaktionen mit dem Gesamtsystem.

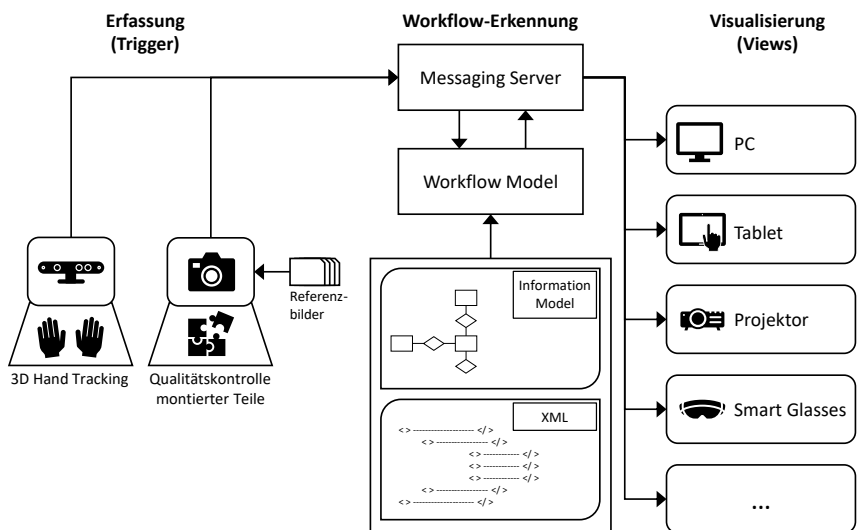


Abbildung 3.6: Systemarchitektur des SmartFactory<sup>KL</sup>-Assistenzsystems (Quelle: [Qu16])

Der rechte Bereich der Abbildung zeigt die *Ausgabe*, in der die Ergebnisse der Verarbeitung visualisiert werden. Beim SmartFactory<sup>KL</sup>-System ist die Ausgabe beziehungsweise die Visualisierung nicht an ein spezifisches Gerät gekoppelt, sondern kann über verschiedene Systeme oder Geräte durchgeführt werden. Das System folgt einem modularen Aufbau und kann an die Bedürfnisse der nutzenden Person und den Arbeitsprozess angepasst werden. Die Komponente des Messaging Servers agiert als Kommunikationsschnittstelle und Verbindungsglied, um alle Erfassungs- und Visualisierungssysteme mit der Workflow-Erkennung (Verarbeitungs-komponente) zu verbinden.

Die Auswertung der Erfassungsbögen zu aktuellen Assistenzsystemen und der Vergleich mit der SmartFactory<sup>KL</sup>-Systemarchitektur in Abbildung 3.6 führt zur Anpassung der Systemarchitektur in Abbildung 3.7. Die zentralen Bereiche der Erfassung, Verarbeitung und Visualisierung bleiben

in der abstrahierten Systemarchitektur enthalten. Änderungen zeigen sich im Bereich der Erfassung und der Prozessverarbeitung. Während bei der *SmartFactory*<sup>KL</sup>-Systemarchitektur fest definierte Erfassungssysteme bestehen, enthält die abstrahierte Architektur einen modularen Erfassungsbereich, bei dem die Sensorsysteme ausgetauscht oder weitere Systeme eingebunden werden können. Die Modularisierung des Eingangsbereiches wird insbesondere aufgrund der verfügbaren Assistenzsysteme mit einem modularen Aufbau durchgeführt. Ein Beispiel ist das „Active Assist“-System<sup>11</sup>.

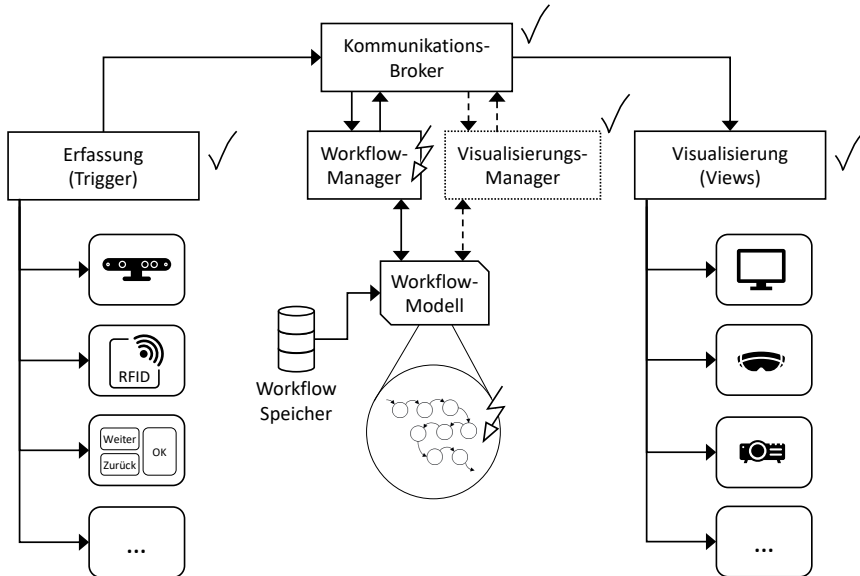


Abbildung 3.7: Abstrahierter Systementwurf in Form einer Systemarchitektur des Assistenzsystems

Eine weitere Anpassung erfolgt durch die Ergänzung des Visualisierungs-Managers im Bereich der Verarbeitung. Dieser repräsentiert die zentrale Rolle der Aufbereitung von Informationen für die nutzgerechte Visualisierung als Unterstützung für den Arbeitsprozess. Der Verarbeitungsfokus des *SmartFactory*<sup>KL</sup>-Systems liegt hingegen auf der Erfassung des Arbeitsprozesses und der Bestimmung des aktuellen Arbeitsschritts innerhalb des Arbeitsauftrages (vgl. Verarbeitungskomponente in Abschnitt 3.4.1). Die Aufgabe der Arbeitsschritterkennung wird in Abgrenzung zu Abbildung 3.6 durch die explizit dargestellte Komponente des Workflow-Managers dargestellt. Zusätzlich enthält der abstrahierte Systementwurf ein Workflow-Modell, in Anlehnung an das Workflow-Modell der *SmartFactory*<sup>KL</sup>-Systemarchitektur.

Die letzte Änderung betrifft die Bereitstellung des Workflow-Modells durch eine Workflow-Datenbank, in der alle verfügbaren Arbeitsprozesse abgespeichert sind. Diese Ergänzung folgt

<sup>11</sup>vgl. Seite 138 in Anhang A

aus den Anforderungen des abstrahierten Komponentenentwurfs (siehe Abschnitt 3.4.1) und den Informationen der Ergänzungsbögen. Die Aufgabenbereiche der Erstellung und der Verwaltung von Workflow-Modellen sowie deren Übertragung in die Workflow-Datenbank liegen außerhalb des Betrachtungsbereiches dieser Arbeit. Die Haken bzw. Blitze an den Komponenten von Abbildung 3.7 spiegeln die Abgrenzung dieser Arbeit auf den Bereich der Verarbeitungskomponente wider (vgl. Abschnitt 3.2). Haken repräsentieren Komponenten oder Bereiche, die im Rahmen der konkreten Entwicklung als verfügbar angesehen und nicht weiter betrachtet werden. Blitze zeigen die Komponenten an, die im Rahmen der Zielsetzung als nicht funktional angesehen werden und im Zuge der Entwicklung eines kontextsensitiven, modularen Assistenzsystems angepasst beziehungsweise ausgetauscht werden. Eine Beschreibung von Entitäten sowie der Aufgabe einer jeden Entität kann der folgenden Übersicht entnommen werden:

**Trigger** sind Sensorsysteme zum Erfassen von Kontextinformationen und zur eventbasierten Bereitstellung von Kontextinformationen zur Klassifikation der Arbeitssituation. Durch die Kombination mehrerer Trigger-Typen kann die Menge an erfassten Kontextinformationen erhöht werden. Beispielsweise kann ein Sensorsystem zum Erfassen der Handposition Kontextinformationen über den Arbeitsort bereitstellen. Ein Wäge-System hingegen kann Kontextinformationen über die Menge an vorhandenen Materialkomponenten liefern. Redundante Trigger-Typen erhöhen die Genauigkeit der Erfassung, indem sie die Information gegenseitig bestätigen oder eine Inkonsistenz der Erfassung aufzeigen.

**Views** sind Visualisierungssysteme zur Darstellung von Supportinformationen für die nutzende Person des Assistenzsystems. Die Bereitstellung der Informationen ist abhängig von den verwendeten Systemen. Bei der Verwendung mehrerer Anzeigeegeräte können diese unabhängig voneinander betrieben werden. Dadurch können die geeignetsten Systeme für den entsprechenden Verwendungsort oder die Vorlieben der nutzenden Person ausgewählt und kombiniert werden.

**Visualisierungs-Manager (optional):** Das System steuert die Informationsbereitstellung der Views auf Basis des aktuellen Arbeitsschritts und der durch das System erfassten Ereignisse beziehungsweise Kontextinformationen. Abhängig vom Fokus des entsprechenden Assistenzsystems ist der Visualisierungs-Manager in den Workflow-Manager integriert (siehe Visualisierungs-Controller in Abschnitt 3.4.1), besteht als eigenständige Systemkomponente (siehe Abbildung 3.7) oder ist nicht vorhanden. Ziele des Systems sind

1. die Anpassung der Informationsdichte,
2. die Anpassung der Art der Informationsdarstellung und
3. die Anpassung der Art der Hilfestellung an die Bedürfnisse und Anforderungen der aktuell nutzenden Person.

**Kommunikations-Broker (Messaging Server):** Das Kommunikationssystem arbeitet als zentraler Verteiler von Nachrichten für alle Systemkomponenten. Die Verteilung der Nachrichten wird eventbasiert durchgeführt (vgl. Anforderung in Abschnitt 3.4.1) und folgt dem „Publish & Subscribe“-Verfahren<sup>12</sup>, bei dem teilnehmende Systemkomponenten die für sie relevanten Nachrichtenthemen abonnieren. Entsprechend der Abonnements verteilt der Kommunikations-Broker die Nachrichten an die Teilnehmer, die an Nachrichten dieses Themas interessiert sind.

**Workflow-Manager (vgl. Abschnitt 3.4.1):** Der Workflow-Manager ist die zentrale Verarbeitungseinheit zur Steuerung des Arbeitsprozesses mit Zugriff auf das Workflow-Modell. Durch den Eingabebereich erfasste Kontextinformationen werden vom Workflow-Manager verwendet, um den erfolgreichen Abschluss eines Arbeitsschritts und die Weicherschaltung zu einem anderen Arbeitsschritt zu bestimmen. Den Wechsel des Arbeitsschritts übermitteln der Workflow-Manager über den Kommunikations-Broker an das Gesamtsystem.

**Workflow-Modell (vgl. Abschnitt 3.4.1):** Der Arbeitsprozess, bestehend aus allen Arbeitsschritten und den dafür notwendigen Tätigkeiten, Werkzeugen und Bauteilen, wird im Workflow-Modell abgebildet. Anhand des Modells bestimmt der Workflow-Manager die Sequenz der Arbeitsschritte und die Bedingung für den Übergang von einem Arbeitsschritt in den nächsten. Zudem stellt das Modell die Daten für den Visualisierungs-Manager bereit, um der nutzenden Person visuelle Unterstützung zu bieten.

Die Herleitung von Anforderungen aus dem abstrahierten Systementwurf an die Entwicklung der kontextsensitiven Verarbeitungslogik erfolgt in Anlehnung an das Schema aus Abschnitt 3.4.1. Die Anforderungen werden aus der Beantwortung der folgenden Fragen abgeleitet:

- In welcher Art und Weise werden Interaktionen zwischen Trigger und Workflow-Manager durchgeführt?
- In welcher Art und Weise werden Interaktionen zwischen View / Visualisierungs-Manager und Workflow-Manager durchgeführt?
- Welche Informationen werden für die Interaktionen übermittelt?

Der Fall eines in den Workflow-Manager integrierten Visualisierungs-Controllers wird aufgrund des Fokus dieser Arbeit nicht weiter betrachtet. Eine Betrachtung erfolgt jedoch in thematisch verwandten Arbeiten (vgl. Abschnitt 2.1.2). Entsprechende Anforderungen, die sich aus einer Interaktion zwischen Workflow-Manger und den weiteren Entitäten des Gesamtsystems ergeben und einen Bezug zum integrierten Visualisierungs-Controller haben, werden aus diesem Grund nicht erhoben.

<sup>12</sup>vgl. [Eu03; Ja09]. Publish und Subscribe ist ein Interaktions- und Kommunikations-Modell, zum Austausch von Nachrichten zwischen entkoppelten Kommunikationsteilnehmern in einer Viele-zu-Viele Kommunikation.



Für die Entwicklung eines kontextsensitiven, modularen Assistenzsystems ergeben sich, entsprechend der Beantwortung der Fragen, die folgenden lokalen, funktionalen Anforderungen aus dem abstrahierten Systementwurf.

- LA6: Die erfassten und bereitgestellten Ereignisse / Kontextinformationen hängen von der eingebundenen Sensorik ab
- LA7: Die eingebundene Sensorik besteht aus unabhängigen Systemen, die ausgetauscht, erweitert oder reduziert werden können
- LA8: Der Kommunikations-Broker übermittelt die Ereignisse / Kontextinformationen an den Workflow-Manager
- LA9: Der Workflow-Manger übermittelt den aktuellen Arbeitsschritt an den Kommunikations-Broker

### 3.4.3 Abstrahierte Systemspezifikation

Aufbauend auf dem abstrahierten Systementwurf aus Abschnitt 3.4.2 und dem abstrahierten Komponentenentwurf der Verarbeitungskomponente aus Abschnitt 3.4.1 wird die übergeordnete Systemspezifikation als abstrahiertes Verhaltensmodell für aktuelle Assistenzsysteme hergeleitet. Basis für das Modell ist weiterhin das *SmartFactory*<sup>KL</sup>-System, das durch die bisher erarbeiteten Anforderungen sowie die Informationen aus den Erfassungsbögen erweitert und angepasst wird. Informationen zum Systemverhalten des *SmartFactory*<sup>KL</sup>-Systems stehen dem Autor aufgrund eigener, praktischer Erfahrung mit dem System zur Verfügung und werden in Form des hier erstellten Modells expliziert. Die Modellierung des Systemverhaltens erfolgt durch die Darstellung als Flussdiagramm. Die Bestandteile / Elemente des Modells entsprechen den Hauptaufgaben des Systems sowie den sequenziellen Zusammenhängen zwischen diesen Hauptaufgaben. Eine Zuordnung der Aufgaben zu einzelnen Systemkomponenten der Systemarchitektur (vgl. Abschnitt 3.4.2) wird in dieser Darstellung nicht durchgeführt, da die repräsentierten Aufgaben durch ein Zusammenspiel mehrerer Komponenten erfolgen und somit keine eindeutige Zuordnung möglich ist. Direkte Verbindungen zu Aspekten aus dem abstrahierten Systementwurf (Abschnitt 3.4.2) oder dem abstrahierten Komponentenentwurf (Abschnitt 3.4.1) werden jedoch bei der Beschreibung des Systemverhaltens aufgezeigt.

Abbildung 3.8 zeigt das entsprechende Modell der Systemspezifikation, das sich in drei funktionale Bereiche untergliedert. Elemente mit durchgehendem Rahmen stellen die notwendigen Elemente für das Systemverhalten dar. Elemente mit gestricheltem Rahmen zeigen optionale Aufgaben, die in Abhängigkeit des spezifischen Systems durchgeführt oder nicht durchgeführt werden. Der erste Bereich betrifft die Funktionen beim Start eines assistenzgestützten Arbeits-

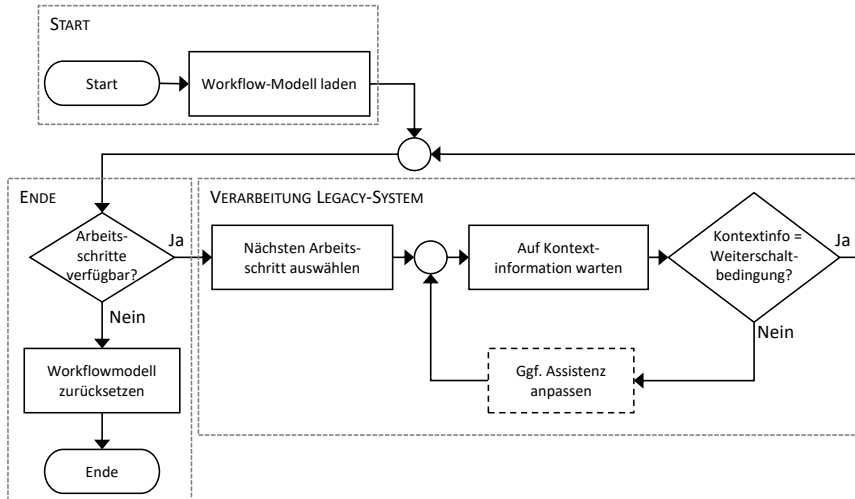


Abbildung 3.8: Abstrahiertes Systemverhalten in Form eines Flussdiagrammes

auftrages (START). In diesem Bereich lädt das System das zum Auftrag gehörende Modell des Arbeitsauftrages. Dieses beinhaltet die notwendigen Arbeitsschritte, die Bedingungen für den Übergang von einem in den nächsten Arbeitsschritt und es wird die Reihenfolge der Arbeitsschritte modelliert. Weitere Informationen zum Workflow-Modell und dessen Aufbau sind in Abschnitt 3.4.1 zu finden.

Im zweiten Bereich findet die eigentliche Unterstützung durch das Assistenzsystem statt (VERARBEITUNG LEGACY-SYSTEM). Dabei wählt das System den nächsten zu bearbeitenden Arbeitsschritt aus. Entsprechend Abschnitt 3.4.1 und 3.4.2 informiert die Verarbeitungskomponente die Visualisierung über den aktuellen Arbeitsschritt, sodass diese eine Unterstützung entsprechend des Arbeitsschritts durchführen kann. In diesem Zustand werden die verfügbaren Informationen und Unterstützungsleistungen zum ausgewählten Arbeitsschritt aufbereitet, sodass eine auf die nutzende Person bezogene Hilfestellung erfolgt. Das System verbleibt in diesem Zustand bis zur erfolgreichen Bearbeitung des aktuellen Arbeitsschritts. Ob ein erfolgreicher Abschluss des Arbeitsschritts stattgefunden hat, entscheidet das System anhand der erfassten Kontextinformationen (vgl. Ereignisse und Kontextinformationen in Abschnitt 3.4.1 und 3.4.2). Zuerst wartet das System auf den Eingang von Kontextinformationen, anschließend prüft das System, ob die erfassten Kontextinformationen mit den Bedingungen für einen erfolgreichen Abschluss aus dem Workflow-Modell übereinstimmen. Stimmen die Informationen überein, schaltet das System weiter und wählt den nächsten Arbeitsschritt aus (vgl. Verarbeitungskomponente in Abschnitt 3.4.1). Falls die Kontextinformationen nicht mit der Bedingung übereinstimmen, verbleibt das System im aktuellen Arbeitsschritt. Da nicht übereinstimmende Kontextinformationen auf den Unterstützungsbedarf der nutzenden Person hinweisen kön-

nen, passt das System gegebenenfalls seine Unterstützung und die bereitgestellten Support-Informationen an die Situation an (vgl. Visualisierungs-Manager in Abschnitt 3.4.2).

Der dritte Bereich des Systemverhaltens zeigt das Ende des Systemprozesses, inklusive Abschluss des assistenzgestützten Arbeitsauftrages (ENDE). Einen Abschluss des Arbeitsprozesses ermittelt das Assistenzsystem durch die Überprüfung der zu bearbeitenden Arbeitsschritte. Sind unbearbeitete Arbeitsschritte verfügbar, wird die Unterstützung im Bereich VERARBEITUNG LEGACY-SYSTEM fortgesetzt. Sind alle Arbeitsschritte erfolgreich bearbeitet und stehen keine weiteren Schritte zur Verfügung, endet die Unterstützung und das Workflow-Modell wird zurückgesetzt.

Um lokale, funktionale Anforderungen des Systemverhaltens für die Entwicklung des kontextsensitiven, modularen Assistenzsystems abzuleiten, wird weiterhin eine Analyse auf Basis von Fragen zum System beziehungsweise Systemverhalten durchgeführt (vgl. Vorgehensweise in Abschnitt 3.4.1 und 3.4.2). Die Anforderungen ergeben sich dementsprechend aus der Beantwortung der folgenden Fragen:

- Welche Informationen nutzt das Assistenzsystem für Entscheidungen und welche Bedeutung haben diese Informationen?
- Woher kommen die Informationen und inwieweit beeinflusst die modulare Bauweise des Systems die Informationsakquise?

Durch die Beantwortung der Fragen zum Systemverhalten leiten sich die folgenden Anforderungen an die Entwicklung einer kontextsensitiven Verarbeitungslogik ab.

LA10: Die zu erfassenden Ereignisse repräsentieren Kontextinformationen, die die aktuelle Arbeitssituation repräsentieren

LA11: Das System muss sicherstellen, dass die im Workflow-Modell erwarteten Kontextinformationen erfasst und bereitgestellt werden können

### 3.4.4 Zusammenfassung lokaler und funktionaler Anforderungen

Das Ziel dieses Kapitels und der an das SRE angelehnten Methodik ist die Aufbereitung von aktuellen Assistenzsystemen und die Herleitung von lokalen, funktionalen Anforderungen für die Entwicklung des kontextsensitiven, modularen Assistenzsystems. Die lokalen, funktionalen Anforderungen gemeinsam mit den globalen Anforderungen (vgl. Abschnitt 3.3) bilden die Grundlage zur Entwicklung des Systems. Um eine allgemeingültige Betrachtung gewährleisten zu können, wurde aufbauend auf dem spezifischen Assistenzsystem der *SmartFactory*<sup>KL</sup> eine Abstrahierung des Systemaufbaus und -verhaltens sowie der Verarbeitungslogik durchgeführt. Hierzu wurden weitere aktuelle Assistenzsysteme betrachtet und deren zentralen Aspekte hinsichtlich Eingangsgröße, Verarbeitungsprinzip und Ausgabegröße analysiert und mittels Erfassungsbo-

gen repräsentiert. Durch den Vergleich des *SmartFactory*<sup>KL</sup>-Assistenzsystems mit den Ergebnissen in den Erfassungsbögen wurde eine Abstrahierung und Anpassung hinsichtlich

- Komponentenentwurf der Verarbeitungseinheit (Abschnitt 3.4.1),
- Systementwurf (Abschnitt 3.4.2) und
- Systemspezifikation / -verhalten (Abschnitt 3.4.3)

durchgeführt, um ein allgemeingültiges Modell für bestehende Assistenzsysteme zu erhalten. Für jeden Teilbereich dieses abstrahierten Modells wurden lokale und funktionale Anforderungen abgeleitet, die sich für die Entwicklung sowie die Integration des kontextsensitiven, modularen Systems ergeben. Die gesammelten Anforderungen finden sich jeweils am Ende der Abschnitte 3.4.1, 3.4.2 und 3.4.3.

## 4 Synthese und Modellbildung

Anschließend an die Informationsakquise und -aufbereitung in Kapitel 3, in der die Anforderungen an die Entwicklung des kontextsensitiven, modularen Assistenzsystems definiert wurden, erfolgt nachfolgend die Entwicklung des Assistenzsystems. Hierzu wird im ersten Schritt das zu erreichende Zielverhalten des Systems modelliert und aufbauend auf den Anforderungen aus Abschnitt 3.3 und 3.4.4 sowie den erarbeiteten Modellen aus Abschnitt 3.4.1 bis 3.4.3 die schrittweise Entwicklung der Verarbeitungskomponente durchgeführt. Der Entwicklungsmethodik (vgl. Abschnitt 3.1) folgend entspricht das Vorgehen den Prinzipien des V-Modells und den Prinzipien „Vom Groben zum Detail“ aus dem Bereich des SE [Hä04; Ha19]. Die Entwicklung und Modellierung des Systems beginnt daher mit einer abstrakten Sicht auf das Gesamtsystemverhalten und wird mit jedem Entwicklungsschritt spezifischer bis hin zu einem Detailentwurf der Verarbeitungskomponente des Systems. Der Aufbau des Kapitels orientiert sich dabei an den drei Entwicklungsstufen des V-Modells. In Abschnitt 4.1 erfolgt die Spezifikation des Zielsystems anhand des geforderten Gesamtsystemverhaltens. Hierbei wird aufbauend auf dem Modell der abstrahierten Systemspezifikation bestehender Assistenzsysteme (vgl. Abschnitt 3.4.3) und den gesammelten Anforderungen ein Modell des Zielsystems erstellt. Anschließend wird in Abschnitt 4.2 die Systemarchitektur entwickelt. Der Fokus des Systementwurfes ist die Erarbeitung der notwendigen Systemkomponenten sowie sich daraus ergebender Anforderungen an die Verarbeitungskomponente. Im letzten Schritt erfolgt die Erstellung des Komponentenentwurfs der Verarbeitungskomponente in Abschnitt 4.3. Aufgrund der Zielsetzung und des Fokus dieser Arbeit unterteilt sich der Komponentenentwurf in die Entwicklung und Modellierung der Verarbeitungslogik und das Prozessmodell (Abschnitt 4.3.2 bis 4.3.4) sowie den Entwurf der Systementitäten, die zur Umsetzung der Verarbeitungslogik notwendig sind (Abschnitt 4.3.5).

### 4.1 Systemspezifikation: Prozessmodell des Zielsystems

Aufbauend auf dem abstrahierten Systemverhalten aktueller Assistenzsysteme (siehe Abschnitt 3.4.3) und der Zielsetzung dieser Arbeit wird das Systemverhalten des angestrebten Systems modelliert. Ziel der Modellierung ist die Herleitung weiterer Anforderungen an die Entwicklung der kontextsensitiven Verarbeitungslogik. Entsprechend des Systemverhaltens bestehender Assistenzsysteme wird wie in Abbildung 3.8 eine Unterteilung in drei funktionelle Bereiche durchgeführt. Die Bereiche START und ENDE des Systemverhaltens werden dabei

für die Modellierung des Zielsystemverhaltens übernommen, da diese keinen Einfluss auf das zentrale Arbeitsverhalten haben. Eine Anpassung des Modells erfolgt daher lediglich im Bereich VERARBEITUNG LEGACY-SYSTEM. Hierbei ist zu beachten, dass aufgrund des Fokus dieser Arbeit im weiteren Verlauf keine detaillierte Betrachtung des Visualisierungsverhaltens durchgeführt wird. Es gilt die Annahme, dass es eine Visualisierungskomponente als Bestandteil der Systemarchitektur (vgl. Abschnitt 3.4.2 und Abschnitt 4.2) gibt, die diese Aufgabe nach denselben Prinzipien und auf derselben Datenbasis durchführt wie dies bei aktuellen Systemen der Fall ist.

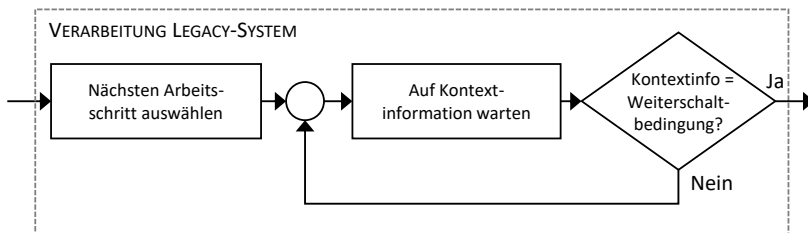


Abbildung 4.1: Ausschnitt für den Bereich VERARBEITUNG LEGACY-SYSTEM des Systemverhaltens bestehender Systeme (vgl. Abbildung 3.8 in Abschnitt 3.4.3)

Im Abgleich des Systemverhaltens aktueller Systeme (siehe Abbildung 4.1) und den globalen Anforderungen zeigt sich, dass Anforderung **GA10** (die Wahrung kausaler Abhängigkeiten) bereits erfüllt wird. Aufgrund des deterministischen Ablaufs, bei dem ein Schritt nach dem anderen bearbeitet wird, werden kausale Abhängigkeiten zwischen den Arbeitsschritten gewahrt. Anforderung **GA9** (Selbstbestimmung der Nutzer) ist aus diesem Grund jedoch nicht erfüllt, da ein Wechsel zwischen Arbeitsschritten nicht möglich ist, ohne den aktuellen Arbeitsschritt fertig bearbeitet zu haben. Zur Berücksichtigung dieser Anforderungen unterscheidet sich das Zielverhalten in zwei Punkten vom Verhalten bestehender Systeme. Einerseits muss für die Erkennung eines Arbeitsschritts, der bearbeitet werden soll, keine feste Reihenfolge durchlaufen werden, andererseits ist der Wechsel von einem Arbeitsschritt zu einem anderen nicht an den erfolgreichen Abschluss des aktuellen Arbeitsschritts gekoppelt. Das sich daraus ergebende Modell des Zielsystemverhaltens ist in Abbildung 4.2 als Flussdiagramm abgebildet.

Im Konkreten bedeutet dies für das Verhalten im Bereich VERARBEITUNG, dass das System zuerst alle Arbeitsschritte bestimmt, die aufgrund der kausalen Abhängigkeiten des Arbeitsprozesses zum aktuellen Zeitpunkt durchführbar sind. Anschließend bewertet das System für jeden dieser Arbeitsschritte die Wahrscheinlichkeit, dass der Nutzer des Systems sich in diesem Arbeitsschritt befindet. Die Bewertung erfolgt dabei auf Basis der Kontextinformationen, die durch das System erfasst werden. Nach Abschluss der Bewertung wird der Arbeitsschritt ausgewählt, der am wahrscheinlichsten durch die nutzende Person bearbeitet wird. Da die erfolgreiche Bearbeitung eines Schritts nicht direkt mit der Erkennung eines Arbeitsschritts gekoppelt ist, wird zusätzlich

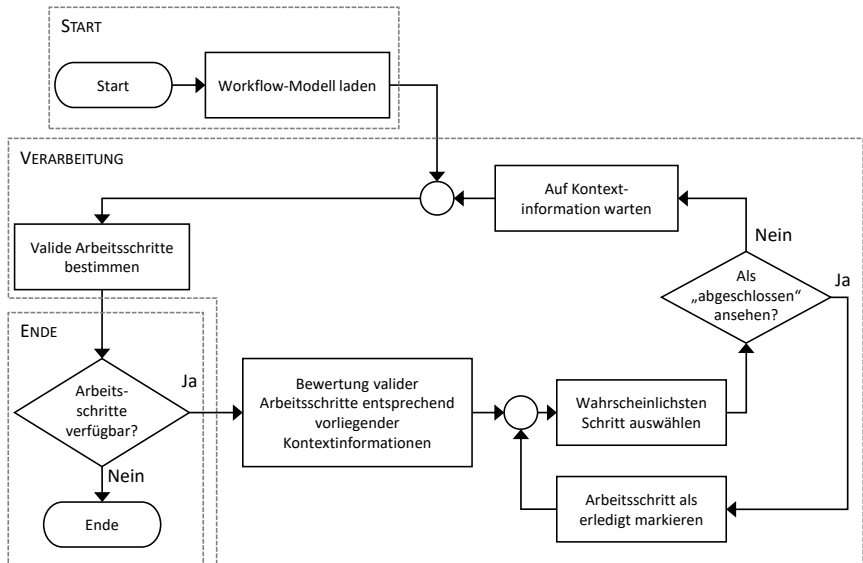


Abbildung 4.2: Modell des Zielsystemverhaltens in Form eines Flussdiagrammes

überprüft, ob der ausgewählte Arbeitsschritt als erfolgreich abgeschlossen betrachtet wird oder nicht. Ist ein erfolgreicher Abschluss des Arbeitsschritts erkannt, wird dieser durch das System als solches markiert. Für die weitere Bearbeitung des Arbeitsprozesses wird dieser Arbeitsschritt nicht mehr berücksichtigt. Wird der Arbeitsschritt nicht als abgeschlossen erkannt, unterstützt das System die nutzenden Person weiterhin bei diesem Arbeitsschritt. In diesem Zustand der Unterstützung verbleibt das System bis zum Eingang von neuen Kontextinformationen. Sobald neue Kontextinformationen erfasst werden, durchläuft das System den gesamten Arbeitsprozess des Bereiches VERARBEITUNG erneut.

Die Anpassung des Systemverhaltens an die Zielsetzung dieser Arbeit ändert das Aufgaben- und Funktionsspektrum des Systems gegenüber dem Verhalten bestehender Systeme. Hierdurch ergeben sich neue lokale, funktionale Anforderungen an die weitere Entwicklung des Systems, die durch die Erfassung und Beschreibung der neuen Funktionalitäten definiert werden. Die dabei gesammelten Anforderungen dienen als Ergänzung der Anforderungen aus Abschnitt 3.4.4. Aus dem veränderten Verhalten ergeben sich die folgenden Anforderungen für die Entwicklung des kontextsensitiven, modularen Assistenzsystems.

- LA12: Das System muss die durchführbaren Arbeitsschritte des aktuellen Arbeitsprozesses bestimmen
- LA13: Das System muss aus den durchführbaren Arbeitsschritten denjenigen auswählen, der am wahrscheinlichsten bearbeitet wird
- LA14: Die Bewertung und Auswahl der Arbeitsschritte erfolgt auf Basis der vorliegenden Kontextinformationen
- LA15: Das System muss den erfolgreichen Abschluss eines Arbeitsschrittes erkennen
- LA16: Bereits abgeschlossene Arbeitsschritte müssen als abgeschlossen markiert werden und dürfen nicht mehr als durchführbar gewertet werden

## 4.2 Systementwurf: Entwicklung der Systemarchitektur

Aufbauend auf den Vorarbeiten und Anforderungen der Systemspezifikation, wird in diesem Kapitel die Entwicklung und Modellierung der Systemarchitektur durchgeführt. Ziel des Systementwurfes ist die Modellierung aller Systemkomponenten sowie der Interaktionen zwischen den Systemkomponenten, um das spezifizierte Systemverhalten zu realisieren. Die detaillierte Ausarbeitung der Funktionsweisen einer jeden Systementität ist kein Bestandteil des Systementwurfes. Der Detailentwurf der Verarbeitungslogik und deren relevanten Aufgaben- und Funktionsbereiche erfolgt im nachfolgenden Abschnitt 4.3.

Die Entwicklung des Systementwurf und der relevanten Entitäten wird unter Berücksichtigung der direkten Anforderungen aus der Systemspezifikation (vgl. Abschnitt 4.1) sowie der gesamten globalen (vgl. Abschnitt 3.3) und lokalen (vgl. Abschnitt 3.4.4) Anforderungen durchgeführt. Hierbei haben insbesondere die folgenden Anforderungen einen Einfluss auf den Aufbau der Systemarchitektur und damit die Entwicklung des Systementwurfs:

- **GA2:** Übertragbarkeit und Wiederverwendbarkeit
- **GA3:** Modifizierbarer Systemaufbau
- **GA4:** Anpassbares Systemverhalten

Um eine Realisierung dieser Anforderungen zu erreichen, werden für die Entwicklung der Systemarchitektur die Prinzipien des Entwicklungsmusters *Separation of Concerns* (SoC) angewendet [Mi90]. Das Ziel dieses Entwicklungsmusters ist die Aufteilung von dedizierten Aufgaben oder Funktionen auf individuelle Systemkomponenten, um einerseits die Komplexität des Systems zu reduzieren und andererseits einen modularen und anpassbaren Systemaufbau zu erhalten.



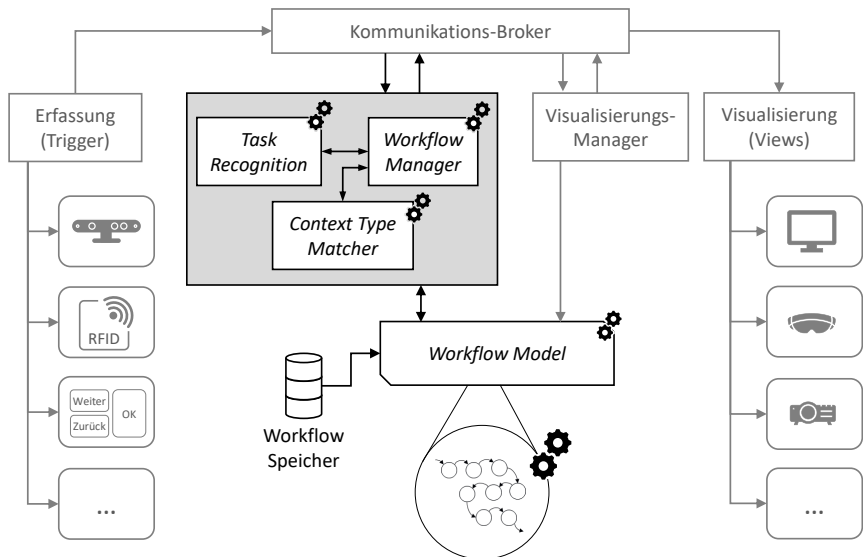


Abbildung 4.3: Systemarchitektur des kontextsensitiven, modularen Assistenzsystems

Die Systemarchitektur, die entsprechend dieses Vorgehens entwickelt wurde, ist in Abbildung 4.3 dargestellt. Die ausgegrauten Entitäten der Architektur repräsentieren die Bereiche der Eingabe (Erfassung), der Ausgabe (Visualisierung) und der Kommunikation, die im Rahmen dieser Arbeit (vgl. Systemabgrenzung in Abschnitt 3.2) nicht weiter betrachtet werden. Die Bestandteile dieser Bereiche wurden aus der Systemarchitektur bestehender Systeme übernommen (vgl. Abschnitt 3.4.2). Interaktionen und Anforderungen, die durch diese Entitäten an die Entwicklung der Verarbeitungskomponente gerichtet sind, werden über die lokalen und funktionalen Anforderungen berücksichtigt. Im Vergleich zur abstrahierten Systemarchitektur bestehender Assistenzsysteme aus Abschnitt 3.4.2 ergibt sich eine Veränderung der Systemarchitektur im Bereich der Verarbeitungskomponente (ehemals Workflow-Manger). Beibehalten wird das Konzept einer zentralen Datenbasis in Form des *Workflow Models* sowie dessen Bereitstellung über eine Workflow-Datenbank. Ferner wird das Prinzip einer einzigen Verarbeitungskomponente übernommen, die mit dem Gesamtsystem und mit dem *Workflow Model* interagiert. Um dennoch die Prinzipien des SoC anzuwenden, erfolgt eine Unterteilung der Verarbeitungskomponente in drei Teilkomponenten. Die Komposition zu einer gemeinsamen Verarbeitungskomponente sichert hierbei die einfache Integration in bestehende Systeme, ohne eine Neuentwicklung in den Bereichen der Erfassung und Visualisierung durchführen zu müssen. Gleichzeitig wird eine einfache Erweiterbarkeit des Funktionsumfangs ermöglicht. Die Funktionen der Teilkomponenten ergeben sich insbesondere aus den funktionalen Anforderungen entsprechend der folgenden Übersicht:

- **LA11:** Das System muss sicherstellen, dass die im *Workflow Model* erwarteten Kontextinformationen erfasst und bereitgestellt werden können  
-> *Context Type Matcher*
- **LA12:** Das System muss die durchführbaren Arbeitsschritte des aktuellen Arbeitsprozesses bestimmen  
-> *Workflow Manager*
- **LA13:** Das System muss aus den durchführbaren Arbeitsschritten denjenigen auswählen, der am wahrscheinlichsten bearbeitet wird  
-> *Task Recognition*

Um unnötige Neuentwicklungen zu vermeiden, wurden bei der Entwicklung der Systemarchitektur und deren Teilkomponenten funktionale Teile der Verarbeitungskomponente übernommen. Dies betrifft insbesondere die Aufgabe der Verwaltung des *Workflow Models* sowie das Auswählen und Weiterschalten von Arbeitsschritten. Eine Übersicht über die Aufgaben der drei Teilkomponenten sowie deren Interaktion miteinander gibt die nachfolgende Beschreibung sowie die schematische Darstellung in Abbildung 4.4.

**Workflow Manager:** Der Workflow Manager steuert das Gesamtsystem auf Basis des *Workflow Models*. Hierzu werden Informationen über den aktuellen Arbeitsschritt an das Gesamtsystem bereitgestellt. Durch das *Workflow Model* ermittelt der *Workflow Manager* die kausalen Zusammenhänge des Arbeitsprozesses und bestimmt welche Arbeitsschritte zum jeweils aktuellen Zeitpunkt durchführbar sind. Erfolgreich bearbeitete Schritte markiert der *Workflow Manager* entsprechend im *Workflow Model*. Somit wird sichergestellt, dass jeder Arbeitsschritt durchgeführt wird. Die Informationen darüber, wann ein Arbeitsschritt erfolgreich bearbeitet wurde, und die Informationen darüber, welcher der aktuell durchführbaren Schritte in Bearbeitung ist, bekommt der *Workflow Manager* von der *Task Recognition* übermittelt. Ob das System betriebsbereit ist, um eine Bearbeitung des Arbeitsprozesses und Erfassung der Arbeitsschritte durchzuführen, meldet der *Context Type Matcher* an den *Workflow Manager*.

**Task Recognition:** Die zentrale Aufgabe der *Task Recognition* ist die Erkennung des Arbeitsschritts, der aktuell bearbeitet wird. Die Informationen darüber, welche Arbeitsschritte zum aktuellen Zeitpunkt durchführbar sind, ermittelt die *Task Recognition* aus dem *Workflow Model*. Eine Bearbeitung des Models wird nicht durchgeführt. Zur Bestimmung welcher Arbeitsschritt derzeit bearbeitet wird und wann ein Arbeitsschritt erfolgreich abgeschlossen ist, nutzt die *Task Recognition* die durch den Bereich der Erfassung bereitgestellten Kontextinformationen. Der jeweils erkannte Arbeitsschritt sowie die erfolgreiche Bearbeitung eines Schritts werden dem *Workflow Manager* übermittelt, um die Steuerung des Arbeitsprozesses zu ermöglichen.

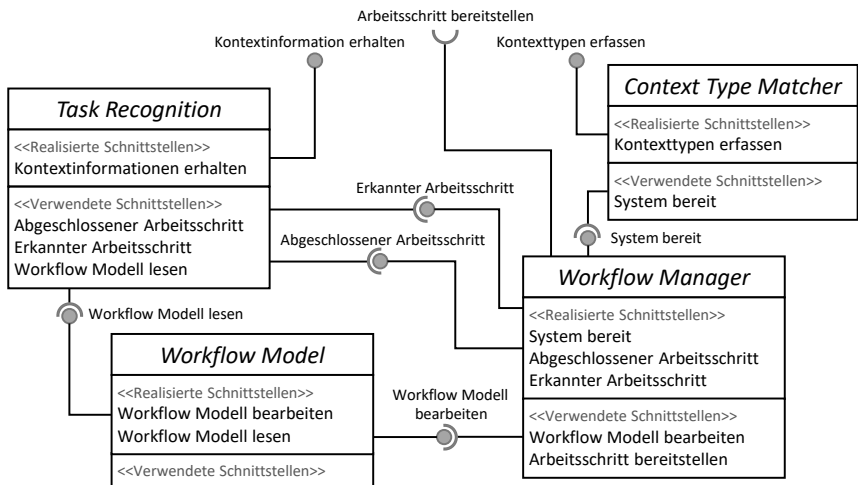


Abbildung 4.4: Verarbeitungsteilkomponenten als UML-Komponentendiagramm

**Context Type Matcher:** Die Arbeitsbereitschaft des Gesamtsystems, um eine Unterstützung für den Arbeitsprozess zu ermöglichen, wird vom *Context Type Matcher* festgestellt. Die Entscheidung darüber trifft der Context-Typ-Matcher durch einen Abgleich der Kontextinformationstypen, die durch die *Task Recognition* erwartet werden, und den Kontextinformationstypen, die durch die Sensorik im Bereich der Eingabe bereitgestellt werden. Sind alle erwarteten Typen durch die Eingabe erfasst, erfolgt die Freigabe der Systembereitschaft.

Die Entwicklung der Funktionslogik der Teilkomponenten erfolgt im Komponentenentwurf der Verarbeitungslogik in Abschnitt 4.3. Die Anforderungen, die sich aus der Systemarchitektur an diese Entwicklung ergeben, werden durch die übergeordneten Aufgaben sowie durch die Interaktionen der Teilkomponenten bestimmt. Dabei mit einzubeziehen sind die Interaktion mit dem *Workflow Model* sowie die Interaktion mit dem Gesamtsystem. Bereits an anderer Stelle betrachtete Anforderungen werden nicht erneut aufgeführt.

- LA17: Die verwendeten Kontextinformationstypen müssen definiert und hergeleitet werden
- LA18: Die Bewertung der durchführbaren Arbeitsschritte auf Basis der eingegangenen Kontextinformationen muss hergeleitet werden
- LA19: Das Workflow-Modell (*Workflow Model*) muss den Arbeitsprozess mit seinen kausalen Zusammenhängen repräsentieren
- LA20: Das Workflow-Modell (*Workflow Model*) muss die Zusammenhänge zwischen Kontextinformationen und Arbeitsschritten repräsentieren

### 4.3 Komponentenentwurf: Entwicklung der Verarbeitungslogik

Nachdem die Entwicklung des Systementwurfes abgeschlossen ist und die notwendigen Komponenten zur Umsetzung des kontextsensitiven, modularen Assistenzsystems hergeleitet sind, folgt der Detailentwurf der Verarbeitungs Komponente und deren Teilkomponenten (vgl. *Workflow Manager*, *Task Recognition* und *Context Type Matcher* in Abschnitt 4.2). Darüber hinaus wird in diesem Kapitel die Entwicklung des *Workflow Models* als zentrale Daten- und Informationsbasis durchgeführt. Hierzu wird im ersten Schritt ein Modell des Arbeitsprozesses hergeleitet, das die Arbeitsprozessfolge und deren kausalen Zusammenhänge repräsentiert (4.3.2). Anschließend erfolgt die Entwicklung der kontextsensitiven Erkennung von Arbeitsschritten mittels semantisch beschriebener Kontextinformationen (4.3.3) sowie die Integration in ein gemeinsames Modell (4.3.4). Grundlage für diese Entwicklung ist die Bewertung bekannter Modelle aus der Automatentheorie und Prozessmodellierung hinsichtlich deren Nutzbarkeit für die Modellierung der zweigeteilten Aufgabe einer Arbeitsschritterkennung und Arbeitsprozesssteuerung (4.3.1). Den Abschluss dieses Kapitels bildet der technische Entwurf der drei Teilkomponenten (*Workflow Manager*, *Task Recognition* und *Context Type Matcher*) sowie des *Workflow Models* (4.3.5).

#### 4.3.1 Bewertung bestehender Modellierungssprachen und Verhaltensmodelle

Die Basis der Arbeitsprozesssteuerung und der Erkennung von Arbeitsschritten ist die zentrale Daten- und Informationsbasis des *Workflow Models*. Bevor eine Entwicklung der entsprechenden Teilfunktionalitäten erfolgen kann, muss der Aufbau und die Funktionsweise des *Workflow Models* definiert werden. Hierzu erfolgt im ersten Schritt eine Bewertung bekannter Modellierungsmethoden aus dem Bereich der Prozessmodellierung und Automatentheorie.

In Abschnitt 2.3 wurden daher die zur Zeit der Erstellung dieser Dissertation gebräuchlichen Modelle vorgestellt, deren Verwendungszwecke und Funktionsweisen beschrieben sowie eine Gegenüberstellung der Modelle (vgl. Abschnitt 2.3.7) durchgeführt. Weiterhin wurde in Abschnitt 2.1.2 gezeigt, dass Assistenzsysteme zur Zeit der Erstellung dieser Arbeit vornehmlich als deterministische Systeme modelliert und umgesetzt werden. Das heißt, die Funktionsweise dieser Systeme kann als endliche Zustandsmaschine angesehen werden, die bei einer eingehenden Folge aus Signalen nach vorherbestimmtem Muster von Zustand zu Zustand schaltet. Eine Alternative zu diesem deterministischen Verhalten bieten Systeme, die auf probabilistischen Modellen aufbauen. Ein Wechsel zwischen Zuständen findet hierbei nicht nach einem vorherbestimmten Muster statt, sondern erfolgt wahrscheinlichkeitsbasiert. Ein Beispiel hierfür liefert das auf HMM basierende Plant@Hand-System (vgl. Abschnitt 2.1.2). In Bezug auf die Entwicklung der kontextsensitiven Verarbeitungslogik zeigt sich jedoch, dass weder ein rein de-

terministisches noch ein rein probabilistisches System geeignet ist, um alle Anforderungen an das Systemverhalten abzudecken. Während das Arbeitsprozessmodell einem festen, vorherbestimmten Prinzip auf Basis der Arbeitsschritte und deren kausalen Zusammenhängen folgt (vgl. **GA10**), existiert kein vorherbestimmtes Verhalten für die Erkennung des Arbeitsschritts, den die nutzende Person als nächstes bearbeiten wird. Ferner besteht die Möglichkeit, dass die nutzende Person ihre Meinung spontan ändert und einen anderen Bearbeitungsweg einschlägt (vgl. **GA9**).

	Nebenläufigkeit	Deterministische Prozesse	Zufallsprozesse	Verborgene Zustände	Ausführbarkeit
EPK	●	●			
BPMN	●	●			●
FSM	●	●	●		●
UML-Zustandsmaschine	●	●			
Petri-Netz	●	●	●		●
HMM			●	●	●

● geeignet    ● teilweise geeignet

Abbildung 4.5: Vergleich von Modellen der Prozessmodellierung und der Automatentheorie mit hervorgehobener Auswahl (Erweiterung von Abbildung 2.29 aus Abschnitt 2.3.7)

Aufbauend auf Abbildung 2.29 aus Abschnitt 2.3.7 stellt Abbildung 4.5 diese Problematik anhand eines Überblicks der untersuchten Modellierungssprachen dar. Im Zuge dieser Untersuchung, unter Betrachtung der Anforderungen an die Verarbeitungskomponente, zeigen sich zwei Systeme, die in Kombination alle Anforderungen erfüllen können. Für den Bereich der kontextsensitiven, zufallsprozessbasierten Modelle sind dies die HMM. Insbesondere zeigt sich deren Möglichkeit zur Betrachtung von verborgenen Zuständen und deren Prinzip mithilfe von Beobachtungen auf diese verborgenen Zustände zu schließen als geeignet<sup>13</sup>. Petri-Netze hingegen sind die einzige Modellierungssprache, die eine direkte Betrachtung von nebenläufigen Prozessen ermöglicht und deren Modelle ausführbar sind. Zwar erfordern die Anforderungen zur Entwicklung der Verarbeitungskomponente keine direkte Verwendung von ausführbaren Modellen, jedoch sind diese im Sinne einer einfachen und automatisierten Erstellung von *Workflow Models* die bessere Wahl für zukünftige Weiterentwicklungen. Aus diesen Gründen werden HMM und Petri-Netze für die weitere Entwicklung der Verarbeitungskomponente und des *Workflow Models* als Modellierungsmethoden betrachtet.

<sup>13</sup>vgl. [AGU14] und deren Arbeiten zur Aktivitätserkennung für Assistenzsysteme

### 4.3.2 Modellierung der Arbeitsprozessfolge

Damit eine Entwicklung des Arbeitsprozessmodells für die Verarbeitungskomponente möglich ist, erfolgt die Einführung eines Beispielprozesses. Anhand dieses Prozesses werden einerseits die kausalen Abhängigkeiten eines Arbeitsprozesses verdeutlicht und andererseits eine Modellierung des Prozesses auf Basis von Petri-Netzen durchgeführt. Die Grundlage des Beispielprozesses bildet ein Legomodell bestehend aus sechs Einzelteilen, welches in Abbildung 4.6 dargestellt ist.

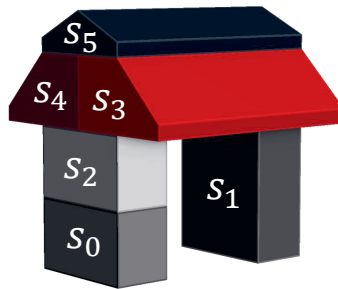


Abbildung 4.6: Arbeitsprozess bestehend aus sechs einzelnen Arbeitsschritten am Beispiel eines Legomodells

Die Notwendigkeit für nebenläufige Prozessfolgen innerhalb eines Arbeitsprozesses zeigt sich in diesem Beispiel anhand der Schrittfolge  $s_0 \rightarrow s_2$  und dem Arbeitsschritt  $s_1$ . Soll das Gesamtmodell gebaut werden, kann zuerst die Schrittfolge und dann der Arbeitsschritt oder erst der Arbeitsschritt und dann die Schrittfolge bearbeitet werden. Ebenso ist es möglich inmitten der Schrittfolgen zum Arbeitsschritt  $s_1$  zu wechseln, da es keine kausalen Abhängigkeiten zwischen diesen gibt. Wird das Legomodell von unten nach oben aufgebaut, so kann die Reihenfolge innerhalb einer Schrittfolge jedoch nicht verändert werden. Schritt  $s_2$  kann somit erst nach Schritt  $s_0$  durchgeführt werden.

Abbildung 4.7 verdeutlicht die Umsetzung einer Modellierung des Beispielsprozesses als Petri-Netz. Die Plätze oder Zustände des Petri-Netzes repräsentieren hierbei die einzelnen Arbeitsschritte des Beispielprozesses. Die Verkettung der Plätze durch Transitionen spiegelt die kausale Abfolge der Arbeitsschritte wider. Diese zeigen sich beispielsweise innerhalb der erwähnten Schrittfolge  $s_0 \rightarrow s_2$ . Für die Nutzung von Petri-Netzen zur Modellierung im Rahmen dieser Arbeit betrachteter Arbeitsprozesse ergeben sich daher Definition 7 und Definition 8.

**Definition 7** Kausal zusammenhängende Arbeitsschritte werden über Transitionen miteinander verkettet, wobei die verbindenden Kanten vom vorgelagerten zum nachfolgenden Arbeitsschritt zeigen.

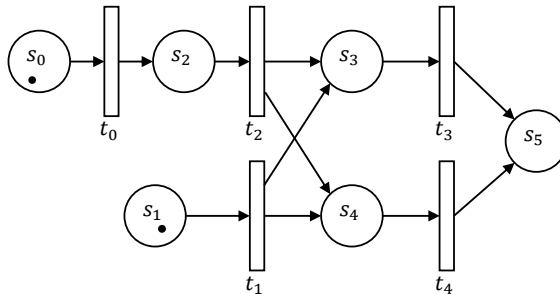


Abbildung 4.7: Initiales Modell eines Petri-Netzes zum Beispielprozess aus Abbildung 4.6

**Definition 8** Voneinander unabhängige Arbeitsschritte oder Schrittfolgen werden durch nebenläufige Pfade modelliert, wobei ...

- ... Arbeitsschritte oder Schrittfolgen ab dem Zeitpunkt eines kausalen Zusammenhanges entsprechend Definition 7 über jeweils eine eigene Transition in einem Arbeitsschritt zusammengeführt werden.
- ... diese entweder in einem Schritt ohne eingehende Kante starten oder durch eine Transition in nebenläufige Pfade aufgeteilt werden (vgl. Abbildung 4.8a).

Entsprechend dieser Modellierung stehen Transitionen für die erfolgreiche Bearbeitung des vorgelagerten Arbeitsschritts. Plätze mit einer Markierung entsprechen den Arbeitsschritten, die zu aktuellem Zeitpunkt durchgeführt werden können. Für Abbildung 4.7 entspricht dies den Startzuständen  $s_0$  und  $s_1$ . Eine Besonderheit für die Modellierung des Prozesses ergibt sich an den Stellen einer Aufteilung oder Vereinigung von nebenläufigen Prozessen (vgl. Transition  $t_1$  und  $t_2$  in Abbildung 4.7). Vereinigungen als auch Aufteilungen von nebenläufigen Prozessen weisen auf eine Veränderung der kausalen Abhängigkeiten im Arbeitsprozess hin. Deutlich wird dies bei den Schritten  $s_3$  und  $s_4$  des Beispielprozesses. Sowohl der Schritt  $s_3$  als auch der Schritt  $s_4$  können erst nach erfolgreicher Bearbeitung aller vorherigen Schritte bearbeitet werden. Um diesen Zusammenhang im Petri-Netz basierten Arbeitsprozessmodell umzusetzen und um ein frühzeitiges Schalten der Transitionen  $t_3$  oder  $t_4$  zu vermeiden, wird eine Anpassung der Kantengewichte entsprechend Abbildung 4.8b durchgeführt. Da alle vorgelagerten Ar-

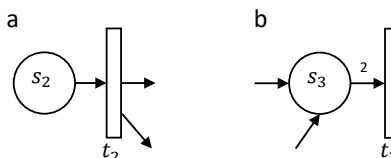


Abbildung 4.8: Besonderheiten in der Modellierung bei der Aufteilung in nebenläufige Prozesse (a) oder dem Vereinigen von nebenläufigen Prozessen (b)

beitsschritte erledigt sein müssen, bedeutet dies für Arbeitsschritt  $s_3$ , dass das Kantengewicht der ausgehenden Kante entsprechend der Anzahl an vorgelagerten Arbeitsschritten gewählt werden muss. Allgemein lässt sich dieser Zusammenhang für alle Situationen entsprechend Definition 9 beschreiben.

**Definition 9** Arbeitsschritte, die von zwei oder mehr Arbeitsschritten abhängig sind, das heißt zwei oder mehr eingehende Kanten besitzen, erhalten eine ausgehende Kante, deren Kantengewicht der Anzahl der eingehenden Kanten entspricht.

Damit eine Bearbeitung aller Arbeitsschritte des Prozesses entsprechend der bisherigen Annahmen möglich ist und das Petri-Netz für die Nutzung als Arbeitsprozessmodell der Verarbeitungskomponente geeignet ist, erfolgt als letzte Anpassung die Erweiterung des Petri-Netzes um einen abschließenden Platz. Dieser Platz repräsentiert das erfolgreiche Bearbeiten aller Arbeitsschritte, sodass alle (nebenläufigen) Prozesse in diesem Platz enden (vgl. Definition 10). Eine grafische Darstellung zur Verdeutlichung dieser Erweiterung ist in Abbildung 4.9a und Abbildung 4.9b zu finden.

**Definition 10** Alle Schritte des Arbeitsprozesses ohne ausgehende Kante werden in einem abschließenden Platz des Petri-Netzes zusammengeführt, wobei jeder Schritt die Verkettung über eine eigene Transition durchführt.

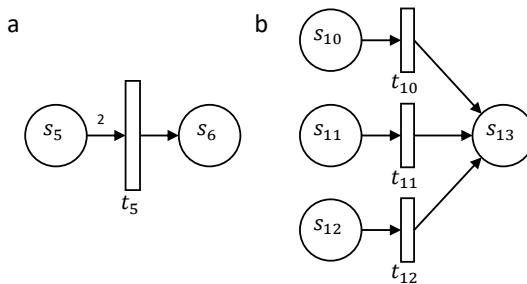


Abbildung 4.9: Modellierung des Endzustandes eines Arbeitsprozesses für den Beispielprozess aus Abbildung 4.6 (a) und beispielhaft für das Ende von drei nebenläufigen Prozessen (b)

Den Modellierungsbedingungen (vgl. Definition 7-10) folgend, lässt sich das Arbeitsprozessmodell des Legoaufbaus entsprechend Abbildung 4.10 darstellen. Im Gegensatz zum initialen Petri-Netz aus Abbildung 4.7 wurden angepasste Kantengewichte vor den Transitionen  $t_3$ ,  $t_4$  und  $t_5$  sowie ein abschließender Platz  $s_6$  als Endzustand hinzugefügt. Das abgebildete Modell befindet sich dabei im Startzustand, indem noch kein Arbeitsschritt bearbeitet wurde. Die Repräsentation des Startzustandes erfolgt dabei über die Markierungen in den Schritten  $s_0$  und  $s_1$ . Hieraus ergibt sich die letzte Modellierungsbedingung zur Umsetzung des Arbeitsprozessmodells mithilfe von Petri-Netzen (vgl. Definition 11).



**Definition 11** Die Anfangsmarkierung ergibt sich aus den Arbeitsschritten ohne Abhängigkeit von einem anderen Schritt, wobei diese Arbeitsschritte als Plätze ohne eingehende Kante dargestellt werden.

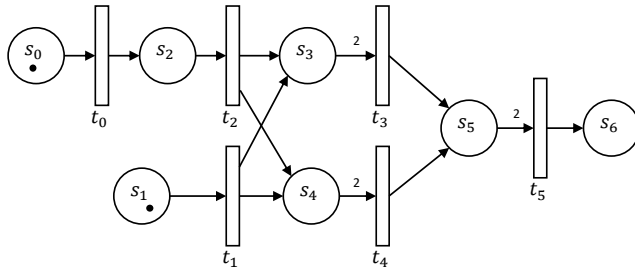


Abbildung 4.10: Vollständiges Arbeitsprozessmodell für den Beispielprozess aus Abbildung 4.6

### 4.3.3 Modellierung der kontextsensitiven Erkennung des Arbeitsschritts

Zusätzlich zur Modellierung des Arbeitsprozesses und der kausalen Zusammenhänge zwischen den Arbeitsschritten muss die Verarbeitungskomponente in der Lage sein, denjenigen Arbeitsschritt auszuwählen, der zum aktuellen Zeitpunkt am wahrscheinlichsten bearbeitet wird (vgl. Abschnitt 4.2 und **LA13**). Die Modellierung dieser kontextsensitiven Erkennung wird in Anlehnung an die HMM (vgl. Abschnitt 2.3.6) durchgeführt, wobei jedoch keine Umsetzung eines HMM erfolgt, sondern Ansätze der Modelle übernommen und auf die Problemstellung dieser Arbeit übertragen werden. Der Grund dafür liegt darin, dass HMM nicht dazu geeignet sind, die kausalen Abhängigkeiten zwischen Arbeitsschritten des Arbeitsprozesses abzubilden (vgl. Abschnitt 4.3.1). Ein zentraler Aspekt der HMM, die Beobachtbarkeit von verborgenen Zuständen, lässt sich jedoch auf die Aufgabe der Erkennung von Arbeitsschritten übertragen. Durch die Erfassung von Kontextinformationen durch das Assistenzsystem stehen Beobachtungen zu einer spezifischen Situation oder einem Zustand zur Verfügung. Der Zustand an sich, das heißt der Arbeitsschritt, der von der nutzenden Person durchgeführt wird, kann nicht direkt beobachtet werden. Stattdessen muss anhand der erfassten Kontextinformationen auf den Zustand zurückgeschlossen werden.

Im Folgenden werden die Aspekte von HMM zur Beobachtbarkeit von verborgenen Zuständen auf die Erkennung von Arbeitsschritten des Arbeitsprozessmodells übertragen. Hierzu werden im ersten Schritt Kontextinformationsklassen hergeleitet, in denen relevante Kontextinformationen eines Typs zusammengefasst werden. Dies liegt daran, dass einerseits nicht alle durch die Sensorik erfassbaren Kontextinformationen betrachtet werden können und dass andererseits nicht alle Kontextinformationen geeignet sind, um einen Rückschluss auf den Arbeitsprozess

zu ermöglichen. Die Herleitung der Kontextinformationsklassen geschieht in Anlehnung an die semantische Modellierung des Arbeitskontextes (vgl. Abschnitt 2.2). Im Speziellen wird auf der Arbeit von ALM UND HADLAK aufgebaut [AH15], in welcher der Kontext durch Beziehungen zwischen der Arbeitssituation, dem Arbeitsort, den beteiligten Personen, den beteiligten Dingen und der Tätigkeit definiert wird (vgl. Abschnitt 2.2.2).

Entsprechend der Abgrenzung aus Abschnitt 3.2 liegt der Fokus dieser Arbeit auf der Erfassung von manuellen Tätigkeiten im Zuge eines Reparaturprozesses, sodass für die weitere Betrachtung folgende Annahmen getroffen werden:

- Die beteiligte Person ist immer ein arbeitende Person
- Die Dinge, die während des Reparaturprozesses benötigt werden, sind Werkzeuge und Materialien / Bauteile (vgl. Abschnitt 2.2.4)
- Die durchgeführten Tätigkeiten stammen aus dem Bereich der manuellen Tätigkeiten (vgl. Abschnitt 2.2.3)
- Der Arbeitsprozess beinhaltet Tätigkeiten an verschiedenen Orten des Arbeitsplatzes, wie einem Arbeitsbereich, einem Lagerbereich oder einem Werkzeugbereich

Aus diesen Annahmen und der Definition von ALM UND HADLAK lassen sich die Kontexttypen „Werkzeug“, „Material“, „Tätigkeit“ und „Ort“ als Kontextinformationsklassen ableiten, die für die weitere Betrachtung der kontextsensitiven Erkennung von Arbeitsschritten verwendet werden. Aufgrund der Zielsetzung dieser Arbeit liegt der Bereich der Erfassung außerhalb des Betrachtungsraumes, sodass keine tiefergehende Untersuchung und Definition aller Kontextinformationen innerhalb dieser Klassen erfolgt.

Nachdem relevante Kontexttypen hergeleitet wurden, wird im nächsten Schritt eine Verknüpfung dieser Typen mit den Arbeitsschritten des Arbeitsprozesses durchgeführt. Hierdurch kann jedem Arbeitsschritt eine Kontextinformation jeden Typs zugewiesen werden. Jede zugewiesene Kontextinformation agiert hierbei als Beobachtung, durch die auf den Arbeitsschritt zurückgeschlossen werden kann. Abbildung 4.11 zeigt diese Verknüpfung von Kontextinformationen und Arbeitsschritten beispielhaft für zwei unterschiedliche Arbeitsschritte. Die Festlegung der spezifischen Kontextinformationen je Arbeitsschritt erfolgt anhand der durchzuführenden Aufgabe des jeweiligen Schritts. Für den im Beispiel dargestellten Schritt  $s_1$  ist die Aufgabe, unter Verwendung des *PH1-Schraubendrehers* eine *Schraube* aus dem Werkstück im *Arbeitsbereich* herauszudrehen. Weitere Materialien oder Komponenten werden bei der Durchführung der Aufgabe nicht benötigt.

Nach diesem Vorbild kann eine semantische Modellierung des Arbeitskontextes für jeden Arbeitsschritt eines Arbeitsprozesses durchgeführt werden. Als Basis für die Modellierung der spezifischen Kontextinformationen dienen bestehende Klassifikationssysteme (vgl. Abschnitt 2.2.4) sowie Normen und Richtlinien für die Beschreibung von manuellen Tätigkeiten (vgl. Abschnitt 2.2.3).

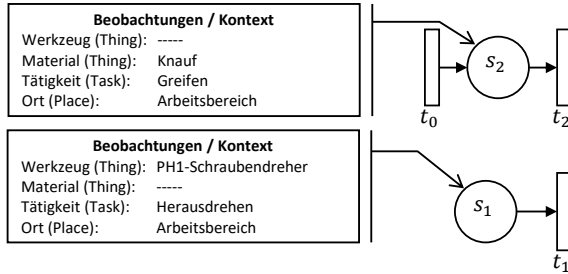


Abbildung 4.11: Verknüpfung von Kontextinformationen mit einem Arbeitsschritt

Die Kombination aus allen spezifischen Kontextinformationen eines Arbeitsschritts wird als optimale Beobachtung dieses Schritts bezeichnet. Für das Beispiel aus Abbildung 4.11 entspricht die optimale Beobachtung für den Schritt  $s_1$  der Kombination der Kontextinformationen: „Werkzeug: PH1-Schraubendreher“, „Tätigkeit: Herausdrehen“ und „Ort: Arbeitsbereich“. Auf Basis dieser optimalen Beobachtung sowie der durch das System gemeldeten Kontextinformationen kann die Verarbeitungsponente einen Rückschluss auf den Arbeitsschritt ziehen, der am wahrscheinlichsten bearbeitet wird. Eine formale Definition der optimalen Beobachtung sowie eine formale Beschreibung dieser Logik kann Definition 12 entnommen werden.

**Definition 12** Sei

$$\vec{b}_t = \begin{pmatrix} \vec{c}_1(t) \\ \vec{c}_2(t) \\ \vec{c}_3(t) \\ \vec{c}_4(t) \end{pmatrix} \text{ eine Beobachtung zum Zeitpunkt } t \text{ und}$$

$$\vec{b}_{s_x} = \begin{pmatrix} \vec{c}_1(s_x) \\ \vec{c}_2(s_x) \\ \vec{c}_3(s_x) \\ \vec{c}_4(s_x) \end{pmatrix} \text{ die optimale Beobachtung des Schritts } s_x$$

wobei gilt

- $c_1$  := Kontextinformation des Typs Werkzeug,
- $c_2$  := Kontextinformation des Typs Material,
- $c_3$  := Kontextinformation des Typs Tätigkeit und
- $c_4$  := Kontextinformation des Typs Ort,

dann entspricht die Wahrscheinlichkeit, dass  $\vec{b}_t$  bei der Durchführung von Schritt  $s_x$  gemacht wurde, dem Grad der Übereinstimmung von  $\vec{b}_t$  und  $\vec{b}_{s_x}$ .

Zur Verdeutlichung der in Definition 12 gezeigten Logik wird die Erkennung eines Arbeitsschritts am Beispiel aus Abbildung 4.11 durchgeführt. Dazu werden verschiedene Beobachtungen, das heißt die Meldungen von unterschiedlichen Kontextinformationen, sowie deren Übereinstimmung mit der optimalen Beobachtung betrachtet. Die Ergebnisse für die Bestimmung des Grades der Übereinstimmung und somit die Erkennung eines Arbeitsschritts für drei Szenarien sind in Tabelle 4.1 dargestellt. Szenario 1 dieser Betrachtung zeigt für beide Arbeitsschritte eine Übereinstimmung bei jeweils zwei von vier Kontextinformationen. Aus Sicht der Verarbeitungslogik entspricht dies einer Übereinstimmung von 50 % und damit einer gleichverteilten Wahrscheinlichkeit, sodass eine eindeutige Erkennung des Arbeitsschritts nicht möglich ist. Die Szenarien 2 und 3 repräsentieren Beispiele, bei denen jeweils eine eindeutige Bestimmung eines Arbeitsschritts möglich ist. Zur Bestimmung des Grades der Übereinstimmung wurde ein Vergleich zwischen jeweils vier gemeldeten Kontextinformationen und vier Kontextinformationen der optimalen Beobachtung durchgeführt. Eine mögliche Alternative hierzu ist die Erweiterung um mehrere Kontextinformationen je Kontexttyp oder die Berücksichtigung von zeitlichen Verläufen in der Erfassung von Kontextinformationen. Bei einer Erweiterung können beispielsweise mehrere Materialien oder Orte erfasst werden, die bei der Bearbeitung eines Arbeitsschritts relevant sind. Die Berücksichtigung von zeitlichen Verläufen würde die Betrachtung von Arbeitssequenzen wie : „Hole den Schraubendreher PH1, drehe die Schraube heraus und lege den Schraubendreher PH1 zurück“, ermöglichen. In Bezug zur Zielsetzung und Abgrenzung dieser Arbeit ist eine Untersuchung der bestmöglichen Bewertungsfunktion nicht vorgesehen. Jedoch erfolgt im Zuge der Umsetzung und Evaluation (vgl. Kapitel 5) eine Untersuchung verschiedener Bewertungsfunktionen, um eine Aussage zur prinzipiellen Verwendbarkeit dieser Erfassungslogik zu ermöglichen.

Tabelle 4.1: Grad der Übereinstimmung zwischen gemachten Beobachtungen und den optimalen Beobachtungen für das Beispiel aus Abbildung 4.11

Szenario	Beobachtung $\vec{b}_t$	Übereinstimmung mit $s_1$	Übereinstimmung mit $s_2$
1	$\begin{pmatrix} \text{---} \\ \text{Abdeckung} \\ \text{Greifen} \\ \text{Arbeitsbereich} \end{pmatrix}$	2 von 4 $\hat{=} 50 \%$	2 von 4 $\hat{=} 50 \%$
2	$\begin{pmatrix} \text{PH1 – Schraubendreher} \\ \text{Abdeckung} \\ \text{Greifen} \\ \text{Arbeitsbereich} \end{pmatrix}$	2 von 4 $\hat{=} 50 \%$	3 von 4 $\hat{=} 75 \%$
3	$\begin{pmatrix} \text{PH1 – Schraubendreher} \\ \text{---} \\ \text{Herausdrehen} \\ \text{Werkzeugbereich} \end{pmatrix}$	4 von 4 $\hat{=} 100 \%$	1 von 4 $\hat{=} 25 \%$

### 4.3.4 Definition des kombinierten Arbeitsprozessmodells

Das *Workflow Model* als zentrale Datenbasis für die Verarbeitungslogik kombiniert die Aspekte einer Prozessmodellierung und der kontextsensitiven Erfassung von Arbeitsschritten. Aus diesem Grund wird aufbauend auf dem Arbeitsprozessmodell aus Abschnitt 4.3.2 und der Arbeitsschritterkennung aus Abschnitt 4.3.3 ein gemeinsames Arbeitsprozessmodell entwickelt, mit dem beide Aspekte abgebildet werden können. Hierzu erfolgt eine Erweiterung des Petri-Netz basierten Modells um zusätzliche Elemente, die eine Verknüpfung der optimalen Beobachtung für jeden Arbeitsschritt des Prozesses ermöglichen. Definition 13 des kombinierten Modells erfolgt in Anlehnung an die formale Definition von Petri-Netzen (vgl. Definition 2 in Abschnitt 2.3.4).

**Definition 13** Das Verarbeitungsmodell ist ein 6-Tupel  $P = (S, T, F, W, m_0, V)$  mit

$S$  einer endliche Menge an **Plätzen** (auch **Zustände** genannt)

$T$  einer endliche Menge an **Transitionen**

$F \subseteq (S \times T) \cup (T \times S)$  der Flussrelation, bestehend aus **Kanten**

$W : F \rightarrow \mathbb{N}$  den **Kantengewichten**

$m_0 : S \rightarrow \mathbb{N}$  der **Anfangsmarkierung**

$\vec{C} \in \mathbb{N}^4 : S$  der **optimalen Beobachtung** (Kombination aus Kontextinformationen)

In Abgrenzung zur Funktionsweise von herkömmlichen Petri-Netzen bedingt die Erweiterung um optimale Beobachtungen und die darauf aufbauende Bestimmung des aktuellen Arbeitsschritts jedoch weitere Anpassungen bezüglich der Funktionsweise des Modells. Entsprechend der Anforderung **LA12** (vgl. Abschnitt 4.1) werden zu jedem Zeitpunkt lediglich die Arbeitsschritte erkannt, die zum jeweiligen Bearbeitungszustand durchführbar sind. Daher werden die durchführbaren Schritte mithilfe des Arbeitsprozessmodells entsprechend Definition 14 und den Erklärungen in Abschnitt 4.3.2 bestimmt.

**Definition 14** „Aktive Zustände“

Ist eine Transition  $t \in T$  aktiv (vgl. Definition 4), dann sind die Zustände des Vorbereiches  $s \in \bullet t$  ebenfalls aktiv.

Abbildung 4.12 zeigt ein kombiniertes Arbeitsprozessmodell entsprechend Definition 13 beispielhaft. Die aktive Transition  $t_7$  und der nach Definition 14 aktive Arbeitsschritt  $s_7$  sind hervorgehoben. Arbeitsschritt  $s_8$  des Modells ist nicht aktiv, da zwei Marken notwendig sind, um die nachfolgende Transition zu schalten (vgl. Definition 4). Zusätzlich zur Hervorhebung der aktiven Zustände und Transitionen verdeutlicht Abbildung 4.12 die Darstellung eines markierten Zustandes. Für das kombinierte Arbeitsprozessmodell wird ein Zustand als markiert bezeichnet, wenn die Arbeitsschritterkennung (vgl. Abschnitt 4.3.3) diesen als denjenigen Arbeitsschritt ermittelt hat, der am wahrscheinlichsten bearbeitet wird (vgl. Definition 15).

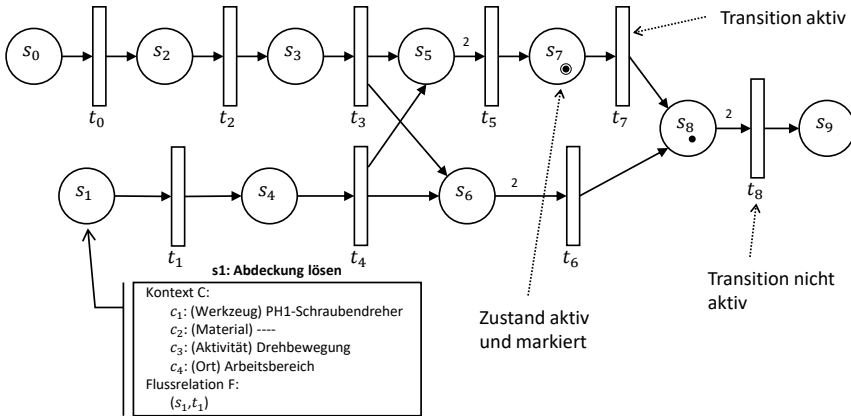


Abbildung 4.12: Beispielhafte Darstellung eines kombinierten Arbeitsprozessmodells der Verarbeitungskomponente

#### Definition 15 „Markierter Zustand“

Wenn  $S^* \subset S$  die Menge aller aktiven Zustände nach Definition 15 ist, dann heißt ein aktiver Zustand  $s \in S^*$  markiert unter der Beobachtung  $B$  wenn gilt  $||C(s) - B|| = \min_{\forall s_x \in S^*} ||C(s_x) - B||$

### 4.3.5 Detailentwurf der Teilkomponenten der Verarbeitungskomponente

Nachdem eine Entwicklung des Arbeitsprozessmodells für die Verarbeitungskomponente durchgeführt wurde (vgl. Abschnitt 4.3.2 - 4.3.4), erfolgt in diesem Schritt die Entwicklung der Teilkomponenten *Workflow Manager*, *Task Recognition* und *Context Type Matcher* sowie des *Workflow Models* zur Repräsentation des kombinierten Arbeitsprozessmodells. Die Herleitung der Detailentwürfe wird auf Basis der übergeordneten Aufgabe der jeweiligen Teilkomponente sowie ihrer Interaktionen mit den anderen Teilkomponenten (vgl. Abschnitt 4.2 und Abbildung 4.4) durchgeführt. Darüberhinaus werden die Anforderungen an die Verarbeitungskomponente berücksichtigt. Hierzu werden im ersten Schritt UML Aktivitätsdiagramme der Teilkomponenten erstellt, die deren Arbeitsweise und Interaktionen modellieren. Die Darstellung von Interaktionen erfolgt durch Sende- und Empfangssignale. Im zweiten Schritt folgt die Modellierung des Aufbaus der Teilkomponenten und des *Workflow Models* mithilfe von UML-Klassendiagrammen.

**Workflow Manager:** Die Aufgaben des *Workflow Managers* unterteilen sich in zwei Tätigkeitsbereiche (vgl. Abbildung 4.13). Der erste Bereich betrifft die Initialisierung eines neuen *Workflow Models* und das Starten einer Arbeitsschritterkennung durch die *Task Recognition*. Zudem behandelt dieser Bereich den Abschluss eines Arbeitsprozesses durch das Zurücksetzen des

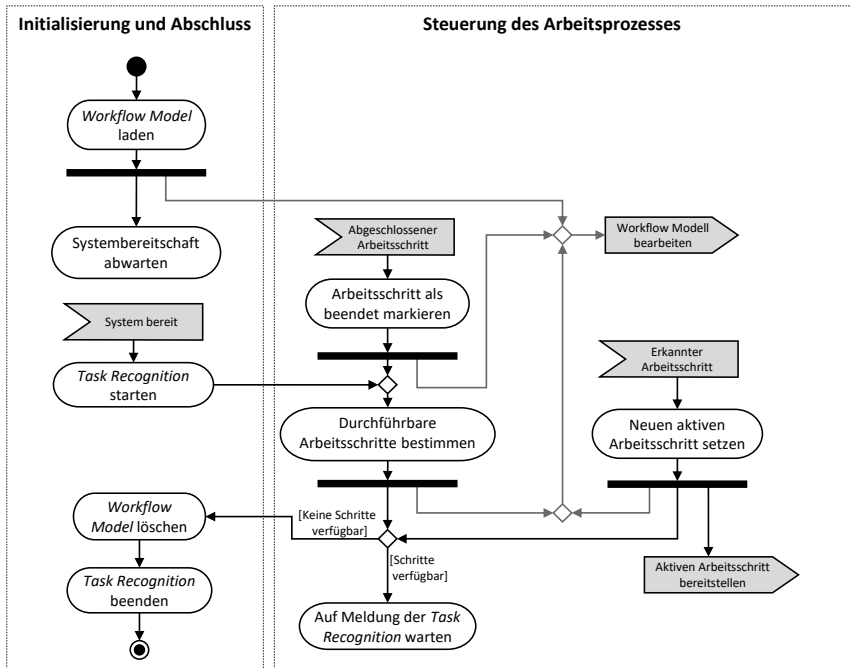


Abbildung 4.13: UML-Aktivitätsdiagramm zur Arbeitsprozesssteuerung durch den *Workflow Manager*

*Workflow Models* und das Stoppen der *Task Recognition*. Im zweiten Bereich findet die eigentliche Steuerung des Arbeitsprozesses (vgl. Abschnitt 4.3.2) und die Interaktion mit der *Task Recognition* und dem Gesamtsystem statt. Die Bedingung für den Übergang in den Bereich der Arbeitsprozesssteuerung ist, dass eine Systembereitschaft durch den *Context Type Matcher* gemeldet wird. Innerhalb der Arbeitsprozesssteuerung durchläuft der *Workflow Manager* iterativ zwei Schleifen, die durch eine Interaktion der *Task Recognition* gestartet werden. Die erste Schleife gewährleistet die Einhaltung der kausalen Zusammenhänge des Arbeitsprozesses, indem die durchführbaren Arbeitsschritte bestimmt werden. Diese Schleife wird beim erfolgreichen Abschluss eines Arbeitsschritts durchlaufen. Die zweite Schleife wird durchlaufen, wenn die *Task Recognition* einen durchführbaren Arbeitsschritt als „in Bearbeitung“ erkennt. In dieser Schleife meldet der *Workflow Manager* den erkannten Arbeitsschritt an das Gesamtsystem und markiert den Schritt im *Workflow Model* als „in Bearbeitung“. Die Bearbeitung des *Workflow Models* wird durch den grau dargestellten Kontrollfluss symbolisiert. Sobald der *Workflow Manager* keine durchführbaren Arbeitsschritte mehr bestimmen kann, gilt die Bearbeitung des Arbeitsprozesses als abgeschlossen.

**Task Recognition:** Die zentrale Aufgabe der *Task Recognition* ist die Erkennung des Arbeitsschritts, der zum aktuellen Zeitpunkt bearbeitet wird. Die Erkennung beruht auf dem Vergleich von erfassten Kontextinformationen und der optimalen Beobachtung der jeweiligen Arbeitsschritte (vgl. Abschnitt 4.3.3). Abbildung 4.14 stellt die Sequenz an Aktivitäten dar, die die *Task Recognition* zur Erkennung eines Arbeitsschritts durchführt. Ein Durchlauf dieser Sequenz wird mit jeder Bereitstellung einer erfassten Kontextinformation initiiert. Während eines Durchlaufs speichert die *Task Recognition* eingehende Kontextinformationen in einer Historie und lädt die Liste an durchführbaren Arbeitsschritten aus dem *Workflow Model*. Anschließend wird für jeden durchführbaren Arbeitsschritt der Grad der Übereinstimmung entsprechend der Kontextinformationshistorie und den optimalen Beobachtungen ermittelt. Der Schritt mit der höchsten Übereinstimmung wird an den *Workflow Manager* übermittelt. Im Falle einer vollständigen Übereinstimmung gilt der Arbeitsschritt als erfolgreich bearbeitet. Der bearbeitete Schritt wird als solcher an den *Workflow Manager* übermittelt und der Arbeitsschritt mit der nächsthöchsten Übereinstimmung wird ausgewählt. Die Arbeitsbereitschaft der *Task Recognition* und die Reaktion auf eingehende Kontextinformationen wird so lange aufrecht erhalten bis die Bearbeitung durch den *Workflow Manager* beendet wird.

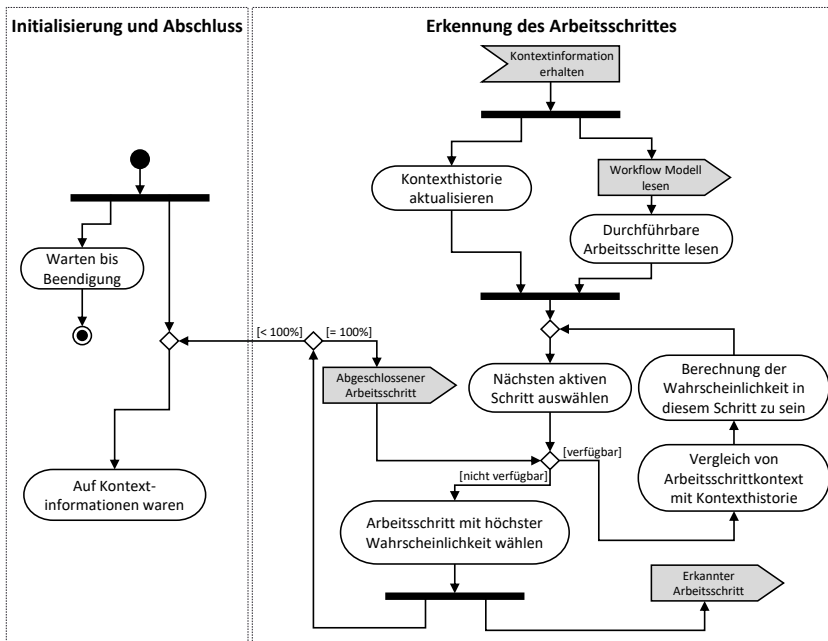


Abbildung 4.14: UML-Aktivitätsdiagramm zur Arbeitsschritterkennung der *Task Recognition*



**Context Type Matcher:** Die Aufgabe des *Context Type Matchers* ist die Überprüfung der bereitgestellten Kontexttypen, um sicherzustellen, dass die verfügbaren Erfassungssysteme alle durch das *Workflow Model* erwarteten Kontexttypen abgedeckt (vgl. LA11). Abbildung 4.15 zeigt das Aktivitätsdiagramm des *Context Type Matchers* zur Umsetzung dieser Aufgabe. Die Überprüfung erfolgt durch ein iteratives Abfragen der Kontexttypen, die durch das System bereitgestellt werden. Wird für jeden der vier definierten Kontexttypen (vgl. Abschnitt 4.3.3) ein System zur Bereitstellung der Kontextinformationstypen ermittelt, wird eine Bereitschaftsmeldung an den *Workflow Manager* übermittelt.

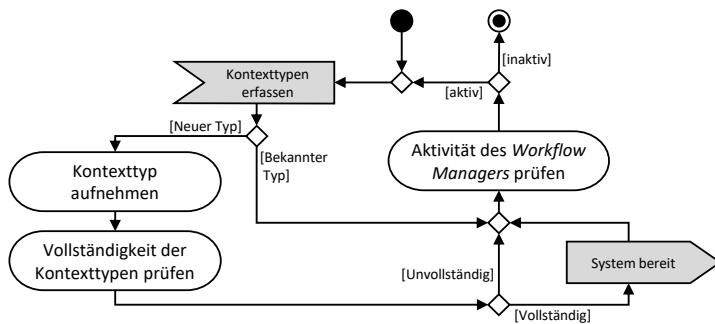
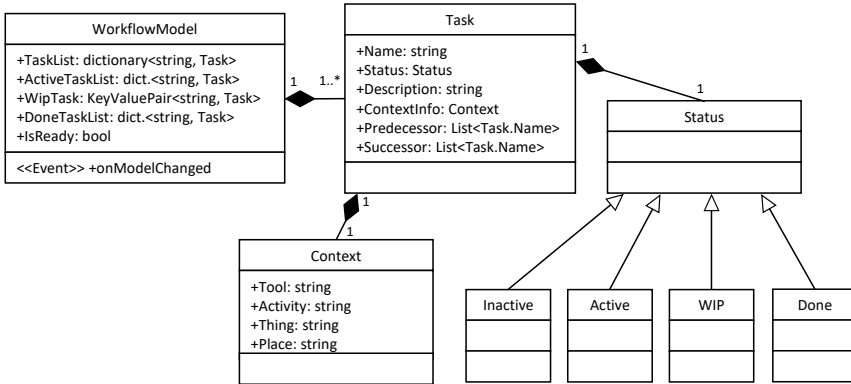


Abbildung 4.15: UML-Aktivitätsdiagramm des *Context Type Matchers*

**Workflow Model:** Das *Workflow Model* repräsentiert den aktuellen Arbeitsauftrag mit allen Arbeitsschritten sowie kausalen Zusammenhängen zwischen Arbeitsschritten und optimalen Beobachtungen zur Identifikation von Arbeitsschritten. Das Modell ist somit die Umsetzung des kombinierten Arbeitsprozessmodells aus Abschnitt 4.3.4 und dient dem *Workflow Manager* und der *Task Recognition* als Daten- und Informationsbasis. Abbildung 4.16 zeigt den Aufbau des *Workflow Models* als UML-Klassendiagramm. Im Detail besteht das Modell aus einer Liste aller Arbeitsschritte (TaskList) sowie aus Teilmengen dieser Liste, die die durchführbaren Arbeitsschritte (ActiveTaskList), die abgeschlossenen Arbeitsschritte (DoneTaskList) sowie den erkannten Arbeitsschritt (WipTask) repräsentieren. Eine Verkettung der Schritte erfolgt über die Verknüpfung eines Arbeitsschritts mit den nachfolgenden Arbeitsschritten (Successor) und den Arbeitsschritten, die vorher bearbeitet werden müssen (Predecessor). Zusätzlich besteht jeder Arbeitsschritt aus einer eindeutigen ID (Name), Beschreibung (Description) und der Modellierung des Kontextes (Context), wobei sich der Kontext aus den vier Kontexttypen „Tool“, „Activity“, „Thing“ und „Place“ zusammensetzt. Der aktuelle Zustand eines Arbeitsschritts wird über den Status des Arbeitsschritts repräsentiert.

Abbildung 4.16: UML-Klassendiagramm des *Workflow Models*

Der Aufbau der Teilkomponenten sowie deren Verbindung zum *Workflow Model* ist als Gesamtmodell in Abbildung 4.17 dargestellt. Das Gesamtmodell folgt dem Entwurfsmuster des Model-View-Controllers, bei dem ein Softwaresystem in die drei Komponenten *Datenmodell*, *Steuerung* und *Präsentation* aufgeteilt wird [De09]. Die Aufgabe der *Steuerung* (*Workflow Manager*) übernimmt alle Veränderungen an den Daten des *Workflow Model* und sorgt dafür, dass die richtigen Daten für die *Präsentation* verwendet werden. Die *Präsentation* (*Task Recognition*) verwendet die bereitgestellten Daten lediglich und hat keinen direkten Zugriff auf das Modell. Änderungen am *Datenmodell* werden der *Präsentation* (*Task Recognition*) nach dem Konzept eines Beobachters mitgeteilt, sodass die diese auf Basis der aktuellen Daten agiert. Zusätzlich reagiert die *Präsentation* auf Nutzereingaben, im Falle der *Task Recognition* sind dies eingehende Kontextinformationen. Veränderungen am Modell, die durch die eingehenden Kontextinformationen ausgelöst werden, werden dem *Workflow Manager* mitgeteilt. Der *Workflow Manager* führt diese Veränderungen dann am *Workflow Model* durch. Durch das Konzept des Model-View-Controllers wird eine lose Kopplung zwischen den Komponenten erreicht. Anpassungen an einer Komponente können durchgeführt werden, ohne dass die anderen Komponenten beeinflusst werden. Zusätzlich kann das System durch zusätzliche *Präsentationen* erweitert werden, die die Daten des *Workflow Models* nutzen, ohne eine Interferenz mit der *Steuerung* oder dem *Datenmodell* zu verursachen.

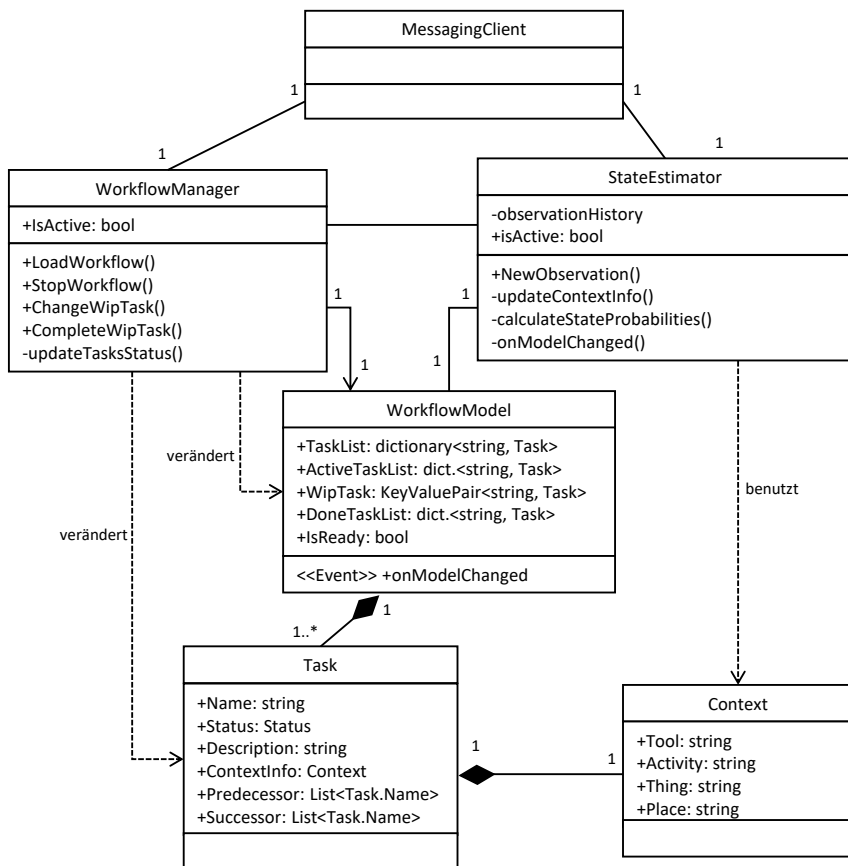


Abbildung 4.17: UML-Klassendiagramm der Teilkomponenten der Verarbeitungskomponente und des *Workflow Models*

## 5 Umsetzung und Evaluation

Nachdem in Kapitel 3 eine Analyse der Anforderungen an ein kontextsensitives, modulares Assistenzsystem und in Kapitel 4 die Entwicklung und Modellierung eines solchen Assistenzsystems erfolgt ist, wird in diesem Kapitel die Umsetzung und anschließende Evaluation des modellierten Systems durchgeführt. Hierzu werden die einzelnen Teilkomponenten der Verarbeitung und das modellierte Workflow-Modell (vgl. Abschnitt 4.3) als Softwaresystem umgesetzt und in den Handarbeitsplatz der *SmartFactory<sup>KL</sup>*, ein bestehendes Assistenzsystem für die Unterstützung bei manuellen Tätigkeiten, integriert. Ferner wird das Assistenzsystem des Handarbeitsplatz so angepasst, dass der Aufbau dem modellierten Systementwurf (vgl. Abschnitt 4.2) entspricht. Um eine anschließende Evaluation des Systems durchführen zu können, geschieht die Umsetzung und Evaluation im Rahmen eines realistischen Anwendungsfalls und mithilfe einer Kombination aus automatisierten und anwenderbezogenen Testszenarien. Entsprechend der gewählten Methodik auf Basis des V-Modells (vgl. Kapitel 3) gliedern sich die Testszenarien in einen Komponententest, einen Integrationstest und einen Systemtest. Den Abschluss der Evaluation bildet eine Prüfung der Zielerreichung.

Der Aufbau des Umsetzungskapitels orientiert sich anhand der Testszenarien, wobei Abschnitt 5.1 mit einer vorgelagerten Beschreibung und Vorstellung des ausgewählten Anwendungsszenarios und des bestehenden Assistenzsystems am Handarbeitsplatz beginnt. Die eigentliche Umsetzung der Verarbeitungskomponente erfolgt zusammen mit der Vorbereitung und Anpassung des bestehenden Assistenzsystems für die Umsetzung in Abschnitt 5.2. Ferner wird die Integration der Verarbeitungskomponente beschrieben. Die Durchführung der ersten Testszenarien im Komponententest erfolgt in Abschnitt 5.3. Hierbei wird die entwickelte Verarbeitungskomponente hinsichtlich ihrer Funktionsweise verifiziert und validiert. Die Durchführung von Testszenarien, zur Prüfung der Integrierbarkeit der Verarbeitungskomponente in das bestehende Assistenzsystem, erfolgt in Abschnitt 5.4. Im anschließenden Systemtest (Abschnitt 5.5) wird das Gesamtsystems und dessen Verhalten mithilfe von Anwendertests untersucht. Den Abschluss dieses Kapitels bildet die Evaluation der Verarbeitungskomponente und deren Funktionsweise auf Basis der in Abschnitt 5.3 - 5.4 durchgeführten Testläufe und der definierten Anforderungen an das kontextsensitive, modulare Assistenzsystem (vgl. Abschnitt 3.2 und 3.4).

## 5.1 Anwendungsfall und Assistenzsystem der SmartFactory<sup>KL</sup>

Als Anwendungsfall zur Evaluation der modellierten Verarbeitungslogik wird die Reparatur eines elektrischen Stellantriebes ausgewählt. Die Aufgabe bei dieser Reparatur ist der Austausch einer defekten Steuerplatine im Inneren des Stellantriebes sowie die Durchführung mehrerer Demontageschritte des Gehäuses, um die Steuerplatine ausbauen und austauschen zu können. Aufgrund des Aufbaus des Stellantriebes stehen dabei unterschiedliche Bearbeitungswege zur Verfügung. Während jedoch ein Teil der Demontageschritte unabhängig voneinander bearbeitet werden kann, existieren auch Demontageschritte, die kausal verkettet sind und nacheinander bearbeitet werden müssen. Nachdem der Ausbau der alten Steuerplatine und der Einbau einer neuen Steuerplatine erfolgt ist, bildet die Remontage der Gehäuseteile den Abschluss der Reparaturaufgabe. Abbildung 5.1 zeigt den Stellantrieb in einem partiell demontierten Zustand. Die blaue Haube des Antriebsgehäuses ist entfernt, wohingegen eine graue Gehäuseabdeckung weiterhin einen Teil der Steuerplatine verdeckt.



Abbildung 5.1: Stellantrieb in teilzerlegtem Zustand

Die notwendigen Arbeitsschritte zum Ausbau der Steuerplatine sind in Tabelle 5.1 aufgelistet. Der Einbau einer neuen Platine und die Remontage des Gehäuses erfolgen entsprechend dieser Liste in umgekehrter Reihenfolge. Aus Zwecken der Übersichtlichkeit ist die Remontage nicht in der Tabelle aufgeführt. Jeder Arbeitsschritt dieses Arbeitsprozesses ist mit einer ID versehen, die den Arbeitsschritt eindeutig identifiziert. Die IDs werden in steigender Reihenfolge durchnummeriert, wobei logisch zusammengehörende Arbeitsschritte unter einer ID zusammengefasst und durch eine zusätzliche Dezimalstelle unterschieden werden. Als logisch zusammenhängend gelten dabei die Arbeitsschritte, die in der angegebenen Reihenfolge und direkt nacheinander bearbeitet werden müssen um eine Veränderung am Werkstück zu erreichen. Anstatt einer Darstellung als logisch zusammenhängende Arbeitsschritte können diese auch als alleinstehender Arbeitsschritt mit geringerem Detailgrad repräsentiert werden. Hierzu zählt insbesondere die Arbeitsschrittkombination ein Werkzeug zu nehmen, dieses zu verwenden und es im Anschluss

wieder zurück zu legen. Zusätzlich zur Übersicht der Arbeitsschritte mit jeweiliger ID zeigt Tabelle 5.1 die kausalen Abhängigkeiten der Arbeitsschritte untereinander. Diese werden in Form von Vorbedingungen dargestellt, die erfüllt sein müssen, bevor der entsprechende Arbeitsschritt bearbeitet werden kann.

Tabelle 5.1: Arbeitsschritte zum Ausbau der Steuerplatine aus dem Stellantrieb

ID	Aufgabe	Vorbedingung
1.1	Innensechskant nehmen	
1.2	Knauf lösen	
1.3	Innensechskant zurücklegen	
2	Knauf nehmen und ablegen	(1) Knauf lösen
3	Haube nehmen und ablegen	(2) Knauf nehmen und ablegen
4.1	Schraubendreher PH1 nehmen	
4.2	Abdeckung lösen	
4.3	Schraubendreher zurücklegen	
5	Abdeckung nehmen und ablegen	(4) Abdeckung lösen
6.1	Schraubendreher Schlitz nehmen	(3) Haube nehmen und ablegen
6.2	Litzen lösen und abziehen	(5) Abdeckung nehmen und ablegen
6.3	Schraubendreher zurücklegen	
7.1	Schraubendreher PH2 nehmen	(3) Haube nehmen und ablegen
7.2	Platine lösen	(5) Abdeckung nehmen und ablegen
7.3	Schraubendreher zurücklegen	
8	Schrauben nehmen und ablegen	(7) Platine lösen
9	Platine nehmen und ablegen	(6) Litzen lösen und abziehen (8) Schrauben nehmen und ablegen

Die Durchführung des dargestellten Reparaturprozesses erfolgt am Handarbeitsplatz der *SmartFactory*<sup>KL</sup>, der mit einem Assistenzsystem zur Unterstützung bei manuellen Tätigkeiten ausgestattet ist (vgl. Abschnitt 2.1.2). Entsprechend der Zielsetzung und Abgrenzung dieser Arbeit (vgl. Abschnitt 3.2) beschränken sich die Veränderungen an diesem System auf den Bereich der Verarbeitung des bestehenden Assistenzsystems (vgl. *Workflow Manager* in Abschnitt 3.4.2). Um die Umsetzung des modellierten Systems und Integration der entwickelten Verarbeitungskomponente durchzuführen, lässt sich eine Anpassung am System nicht vermeiden. Eine detaillierte Beschreibung der Anpassungen und der Vorbereitung des Systems erfolgt im Zuge der Umsetzung in Abschnitt 5.2. Die bereits vorhandenen Systeme zur Erfas-

sung (Eingabe) und zur Visualisierung (Ausgabe) bleiben in ihrer Implementierungsform bestehen und werden weiterverwendet. Abbildung 5.2 zeigt den Aufbau des Handarbeitsplatzes in der zu Beginn der Arbeit bestehenden Form. Zusätzlich werden die verwendeten Bestandteile des Assistenzsystems präsentiert. Im Bereich der Erfassung verfügt das System über eine 3D-Tiefenbildkamera. Der Bereich der Visualisierung und Informationsbereitstellung ist mit einem Monitor für digitale Arbeitsanweisungen sowie einem Projektor zur Projektion von Hinweisen auf die Arbeitsfläche des Arbeitsplatzes ausgestattet.

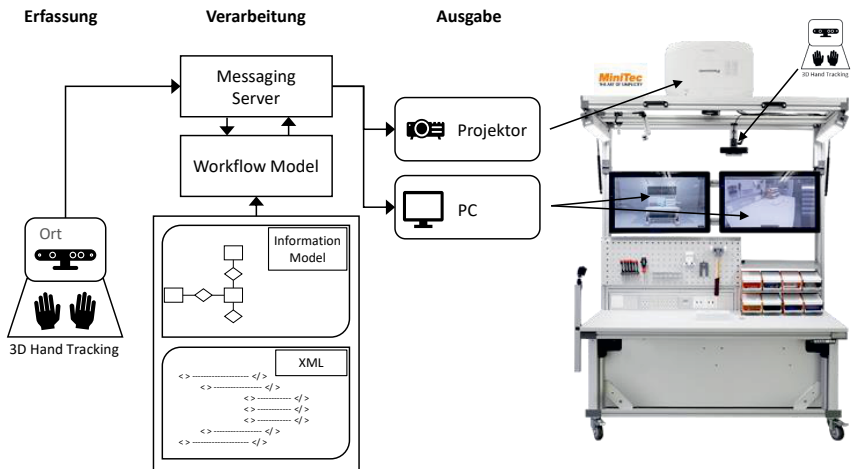


Abbildung 5.2: Aufbau und Bestandteile des bestehenden Handarbeitsplatzes vor der Umsetzung

### Systembestandteile im Bereich der Erfassung

Die Erfassung der Arbeitssituation durch die 3D-Tiefenbildkamera erfolgt durch eine Detektion der Hände der nutzenden Person. Um eine praktische Nutzung dieses Trackings im Assistenzsystem zu erreichen wird der dreidimensionale Arbeitsraum des Arbeitsplatzes in verschiedene Bereiche, wie einen Arbeitsbereich oder einen Werkzeugbereich, aufgeteilt und jeder Bereich mit einer Bezeichnung versehen. Erfasst die Tiefenbildkamera eine Hand der nutzenden Person innerhalb eines definierten Bereiches, so wird der erfasste Bereich an das Gesamtsystem übermittelt. Dadurch erkennt das System, sobald Interaktionen mit Aufbewahrungsboxen, mit Werkzeugen oder anderen Objekten stattfinden. Je nach Anwendungsfall und Szenario können sich die definierten Bereiche im Arbeitsraum unterscheiden und somit andere Informationen während der Bearbeitung eines Arbeitsprozesses gesammelt werden. Beispielsweise können ein einziger Bereich über die Gesamtheit der Werkzeuge oder aber viele Einzelbereiche für die unterschiedlichen Werkzeuge festgelegt werden. Während das erste Setup wenige Detailinformationen liefert wodurch eine Erkennung von verschiedenen Arbeitsschritten nicht möglich ist, birgt das zweite Setup ein erhöhtes Risiko von ungenauen oder fehlerhaften

Erkennungen durch falsch oder zu nah beieinander positionierter Werkzeuge. Die Übermittlung der Information, mit welchen der definierten Bereiche eine Interaktion erkannt wurde, erfolgt an den Kommunikations-Broker (Messaging Server) und findet eventbasiert zum Zeitpunkt der Erfassung statt. Der Inhalt der Übermittlung entspricht der Bezeichnung des erkannten Bereiches. Im weiteren Verlauf der Umsetzung werden die durch die Tiefenbildkamera erfassten Informationen als Kontextinformationen des Kontexttyps „Ort“ verwendet (vgl. Kontexttypen in Abschnitt 4.3.3).

### Systemkomponenten im Bereich der Ausgabe

Die visuelle Unterstützung durch die monitorbasierte Bereitstellung von digitalen Arbeitsanweisungen, präsentiert der nutzenden Person ein Bild des Arbeitsplatzes und des aktuellen Zustandes des Stellantriebes. Das heißt, wenn im aktuellen Arbeitsschritt die blaue Haube demontiert wird, zeigt der Monitor ein Bild des Stellantriebes (ohne Haube) sowie eine Animation der Demontage der Haube (vgl. Abbildung 5.3a). Ferner wird eine textuelle Aufgabenbeschreibung des aktuellen Arbeitsschritts dargestellt sowie eine Kompaktübersicht aller Arbeitsschritte des Reparaturauftrages angezeigt (vgl. rechte Seite von Abbildung 5.3a). Ein Umschalten zwischen den Arbeitsschritten eines Auftrages wird durch die Verarbeitungskomponente des Systems (vgl. *Workflow Manager* in Abschnitt 3.4.2) realisiert.



Abbildung 5.3: Visuelle Unterstützung durch monitorbasierte Arbeitsanweisungen (a) und durch eine In-Situ-Projektion (b)

Im Gegensatz zur monitorbasierten Unterstützung bietet die In-Situ-Projektion eine visuelle Unterstützung direkt im Arbeits- und Tätigkeitsbereich der nutzenden Person. Bei der Durchführung einer Aufgabe kann dadurch eine farbliche Hervorhebung von relevanten Stellen des Stellantriebes oder einer Ablageposition von Materialien und Werkzeugen (vgl. Abbildung 5.3b) entsprechend der Aufgabe des Arbeitsschritts erreicht werden. Ein Umschalten der visuellen Unterstützung geschieht wie bei der monitorbasierten Visualisierung durch die Verarbeitungskomponente.



### Messaging Server / Kommunikations-Broker

Das zentrale Verbindungselement aller Komponenten aus den Bereichen Erfassung, Verarbeitung und Ausgabe, der Messaging Server des Handarbeitsplatzes, arbeitet auf Basis des offenen Nachrichtenprotokolls MQTT (Message Queuing Telemetry Transport) [MQTT]. Das MQTT-Protokoll ermöglicht eine Maschine-zu-Maschine-Kommunikation mittels des Publish und Subscribe-Verfahrens. Der Aufbau der MQTT-Nachrichten teilt sich in ein Informationsthema (Topic), anhand dessen Nachrichten gruppiert werden können, sowie den Nachrichteninhalte (Payload). Die zentrale Stelle eines MQTT-Netzwerks ist der MQTT-Broker, bei dem sich Teilnehmer wie Sensoren oder Softwarekomponenten anmelden können. Verbundene Teilnehmer können einerseits eigene Nachrichten unter einem Topic veröffentlichen, andererseits können diese aber auch einzelne Nachrichtenthemen abonnieren. Die Gruppierung von Nachrichten anhand des Topics macht es für die Teilnehmer unerheblich, welche anderen Teilnehmer im MQTT-Netzwerk angemeldet sind, beziehungsweise welcher Teilnehmer eine Nachricht bereitstellt. Wird ein spezifisches Topic abonniert, so verteilt der MQTT-Broker alle Nachrichten dieses Themas an die jeweiligen Abonnenten.

Im bestehenden Assistenzsystem des Handarbeitsplatzes findet eine Kommunikation mithilfe von zwei Topics statt. Unter dem ersten Topic („HAP/Trigger“) werden alle Nachrichten der Erfassungskomponenten zusammengefasst. Detektiert das Handtracking beispielsweise eine Interaktion im definierten Werkzeugbereich so wird eine Nachricht mit dem Inhalt „Werkzeugbereich“ unter dem Topic „HAP/Trigger“ an den Kommunikations-Broker versendet. Das zweite Topic („HAP/View“) behandelt die Steuerung der Ausgabekomponenten. Im Gegensatz zum ersten Nachrichtentyp beinhalten die Nachrichten an die Ausgabekomponenten jedoch nicht nur eine Information, sondern geben den Ausgabekomponenten beispielsweise an welche Arbeitsanleitung geladen werden soll oder in welchem Arbeitsschritt sich die Anleitung befindet. Darüberhinaus können weitere Informationen wie Hinweistexte zur Darstellung für die nutzende Person übertragen werden. Im Zuge der Umsetzung wird keine Veränderung im Bereich der Ausgabe oder der Steuerung der Ausgabekomponenten durchgeführt (vgl. Abschnitt 5.2) weshalb auf die unterschiedlichen Nachrichteninhalte nicht näher eingegangen wird.

## 5.2 Umsetzung und Integration der Verarbeitungskomponente

Die Umsetzung der Verarbeitungskomponente und des Workflow-Modells für das dargestellte Anwendungsszenario unterteilt sich in drei Phasen. In der ersten Phase wird das bestehende Assistenzsystem für die Integration der entwickelten Verarbeitungskomponente und das Anwendungsszenario vorbereitet. Dies beinhaltet die Außerbetriebnahme der bestehenden Verarbeitungskomponente (Abschnitt 5.2.1), um Interferenzen mit der neuentwickelten Komponente zu verhindern. In der zweiten Phase wird der Arbeitsprozess des Anwendungsszenarios (vgl. Abschnitt 5.1) für die Nutzung durch das Assistenzsystem vorbereitet. Ferner wird das Workflow-

Modell für diesen Arbeitsprozess entsprechend Abschnitt 4.3.5 hergeleitet (Abschnitt 5.2.2). In der dritten Phase erfolgt die eigentliche Umsetzung der Verarbeitungskomponente sowie die Integration der Verarbeitungskomponente in das bestehende Assistenzsystem (Abschnitt 5.2.3).

### 5.2.1 Vorbereitung des bestehenden Assistenzsystems

Die Implementierung der Verarbeitungslogik wird auf Basis der UML-Klassendiagramme und der UML-Aktivitätsdiagramme aus Abschnitt 4.3.5 durchgeführt. Um eine allgemeingültige Umsetzung zu realisieren und zu demonstrieren, dass das entwickelte System in bestehende Assistenzsysteme integriert werden kann, werden die bestehenden Subsysteme des *SmartFactory*<sup>KL</sup>-Assistenzsystems übernommen. Veränderungen an diesen Systemen sind auf ein Minimum reduziert, um eine einfache Integration und Nutzbarkeit der Verarbeitungskomponente zu erreichen. Abbildung 5.4 stellt die Komposition der Systemkomponenten für die Umsetzung im Rahmen dieser Arbeit dar. Der umrahmte Bereich mit hellgrauem Hintergrund repräsentiert hierbei die Softwarekomponenten, bei denen entweder eine Veränderung durchgeführt wird oder eine initiale Implementierung erfolgt. Im Vergleich mit dem bestehenden Aufbau (siehe Abbildung 5.2 in Abschnitt 5.1) zeigt sich, dass eine Anpassung bestehender Komponenten lediglich im Bereich des bisherigen *Workflow Managers* stattfindet. Für die weitere Umsetzung übernimmt der *Workflow Manager* die Aufgabe einer reinen Visualisierungssteuerung und wird daher zukünftig als *Workflow-Visualisierungs-Controller (WVC)* bezeichnet. Durch diese verän-

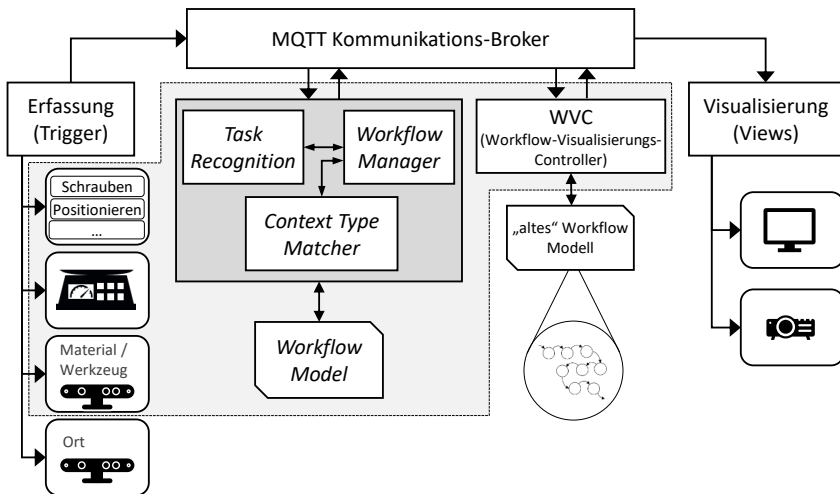


Abbildung 5.4: Schematische Darstellung des Systemaufbaus mit Hervorhebung der angepassten oder ergänzten Systemkomponenten in hellgrau

derte Nutzung wird die bestehende Verarbeitungskomponente außer Betrieb genommen und es muss kein eigenständiger oder neuer Visualisierungs-Controller umgesetzt werden.

Ferner zeigt der Vergleich zwischen dem geplanten und dem bestehenden Aufbau, dass der Bereich der Erfassung um drei Komponenten erweitert wird. Da am Handarbeitsplatz der *SmartFactory*<sup>KL</sup> keine geeignete und funktionsfähige Erkennung von verwendeten Materialien, Werkzeugen sowie von Arbeitsgesten oder Aktivitäten verfügbar ist, werden diese Funktionen durch die Erweiterung ergänzt. Ein Erkennung der genannten Kontextinformationen ist notwendig um die entwickelte Verarbeitungskomponente im Kontext des Anwendungsszenarios verwenden und testen zu können (vgl. Modellierung der kontextsensitiven Arbeitsschritterkennung in Abschnitt 4.3.3). Im Konkreten handelt es sich bei den ergänzten Komponenten um:

- eine intelligente Materialbereitstellung zur Erfassung verwendeter Materialien und Bauteile
- eine optische Objekterkennung zur Erfassung verwendeter Werkzeuge und großer Bauteile
- eine Benutzerschnittstelle zur händischen Eingabe nicht automatisiert erfassbarer Kontextinformationen (z. B. Tätigkeiten und Arbeitsgesten)

Die nicht hervorgehobenen, aber dargestellten Systemkomponenten in den Bereichen der Erfassung und der Ausgabe werden nicht verändert und repräsentieren den Aufbau des Handarbeitsplatzes entsprechend Abbildung 5.2 in Abschnitt 5.1. Im Nachfolgenden werden die Erweiterungen und Anpassungen der Einzelkomponenten näher erläutert. Anschließend werden das entstandene Gesamtsystem und der Versuchsaufbau für die Evaluation präsentiert.

### **Intelligente Materialbereitstellung**

Die intelligente Materialbereitstellung mittels Wägezellensystem ergänzt die Kontextinformationserfassung des bestehenden Assistenzsystems um eine Identifikation der verwendeten Materialien und Komponenten im Laufe des Arbeitsprozesses [Be19b]. Mithilfe der Wägezellen kann die Materialbereitstellung ermitteln, wann und ob Komponenten aus einer der Ablageboxen entnommen oder in diese hineingelegt wurden. Hierzu werden unter jeder am Arbeitsplatz verfügbaren Ablageboxen eine eigene Wägezelle und eine RFID-Lesegerät positioniert. Da die einzelnen Ablageboxen jeweils für spezifische Bauteile vorgesehen sind, können diese mit einem RFID-Tag versehen und durch die RFID-Lesegeräte identifiziert werden (vgl. Abbildung 5.5). Dieser Aufbau ermöglicht es der intelligenten Materialbereitstellung nicht nur Veränderungen des Gewichtes einer spezifischen Ablagebox zu ermitteln, sondern auch die quantitative Mengenänderung bei einer Entnahme- oder Ablageaktivität zu bestimmen. Hierzu werden die Gewichtsveränderungen je Box kontinuierlich überwacht und mit den Stückgewichten der Komponenten der jeweiligen Box verglichen. Welches Stückgewicht welcher Ablagebox zugewiesen ist, bestimmt das System wiederum anhand der RFID-Informationen des RFID-Tags. Wird eine Entnahme- oder Ablageaktivität durch das System erfasst, meldet

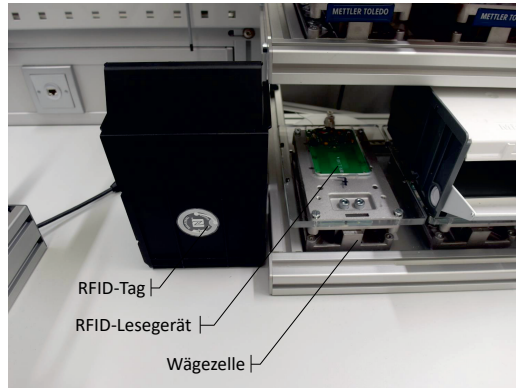


Abbildung 5.5: Intelligente Materialbereitstellung bestehend aus Wägezelle und RFID basierter Materialidentifikation

dieses den Komponententyp und die quantitative Menge an das Gesamtsystem. Im Rahmen der Umsetzung des Anwendungsszenarios werden alle Komponenten des Stellantriebes mit Ausnahme der Steuerplatine und der blauen Haube in Ablageboxen zwischengelagert und somit durch die intelligente Materialbereitstellung erfasst. Aufgrund der Größe von Platine und Haube besitzen diese einen gesonderten Ablageort. Für die Umsetzung werden die von der Materialbereitstellung erfassten Informationen als Kontextinformationen des Kontexttyps „Material“ verwendet.

### Objekt- / Werkzeugerkennung

Die optische Objekt- und Werkzeugerkennung bildet eine Ergänzung des bereits bestehenden Handtracking-Systems. Statt Informationen über den Ort einer Tätigkeit liefert die Objekterkennung Informationen zu den Werkzeugen oder Gegenständen mit denen die nutzende Person direkt interagiert. Das Kernelement dieser Komponente ist wiederum eine Tiefenbildkamera, die die Hände der nutzenden Person detektiert. Darüber hinaus wird jedoch überprüft ob die nutzende Person einen Gegenstand in der Hand hält. Ist dies der Fall bestimmt die Objekterkennung um welchen Gegenstand es sich handelt bzw. ob der Gegenstand dem System bekannt ist. Wird eine Interaktion mit einem bekannten Gegenstand erfasst übermittelt die Objekterkennung dessen Bezeichnung an das Gesamtsystem. Die Umsetzung der beschriebenen Objekterkennung ist im Rahmen des Forschungsprojektes DARAKA<sup>14</sup> durch das Unternehmen CANCONTROLS GMBH durchgeführt worden. Die Softwarekomponente ist als hersteller- und plattformunabhängiges System realisiert und die Übermittlung der Informationen erfolgt eventbasiert. Abbildung 5.6 zeigt einen Ausschnitt der aktiven

<sup>14</sup>DAKARA steht für „Design und Anwendung einer kompakten, energieeffizienten und konfigurierbaren Kameramatrix zur räumlichen Analyse“ und wurde vom Bundesministerium für Bildung und Forschung (BMBF) gefördert. vgl. [DFKI 17; @SF-KL 19]

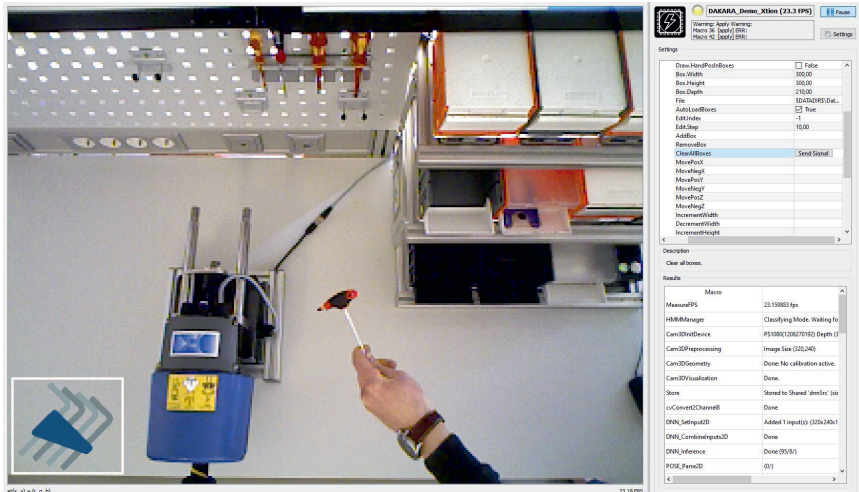


Abbildung 5.6: Aktive Softwarekomponente der Objekterkennung während der Erfassung eines Innensechskant-Schlüssels

Software während die Interaktion mit einem Innensechskant-Schlüssel detektiert wurde. Das erkannte Werkzeug wird durch das weiß umrahmte Symbol in der linken unteren Ecke visuell dargestellt. Im weiteren Verlauf der Umsetzung werden die durch die Objekterkennung bereitgestellten Informationen als Kontextinformationen des Kontexttyps „Werkzeug“ verwendet.

### Vorbereitung der Benutzerschnittstelle (GUI)

Die dritte Erweiterungskomponente für den Bereich der Erfassung ist eine Benutzerschnittstelle zur händischen Erfassung von Kontextinformationen des Kontexttyps „Aktivität“. Im Zuge der Evaluation und der Testphasen wird diese Schnittstelle von einer die Evaluation betreuenden Person verwendet, um Aktivitäten oder Arbeitsgesten zu erfassen und an das Gesamtsystem zu übermitteln. Die nutzende Person selbst hat keinen Zugang zur Benutzerschnittstelle, sondern wird von der betreuenden Person beobachtet, welche somit als System zur Arbeitsgestenerfassung agiert.

Die Umsetzung der Benutzerschnittstelle erfolgt auf Basis eines Dashboards, über das die einzelnen Kontextinformationen mittels verschiedener Buttons ausgewählt werden können. Sobald einer der Buttons betätigt wird, übermittelt die Schnittstelle eine Nachricht mit den Kontextinformationen entsprechend des gedrückten Buttons. Abbildung 5.7 zeigt das umgesetzte Dashboard, wobei sich nicht nur auf die notwendigen Kontextinformationen beschränkt wurde, sondern auch Buttons zur Erfassung aller anderen Kontexttypen sowie zur Steuerung des Systems (vgl. Abschnitt 5.2.3) eingebunden wurden. Zwingend erforderlich für die Nutzung und Evaluation der Verarbeitungskomponente ist die händische Erfassung

Home		
Tool	Place	Material
SCHRAUBENDREHER PH1	LAGERBEREICH	HAUBE
INNENSECHSKANT 4mm	ABLAGEBEREICH	M3x10 RUNDKOPF KREUZSCHLITZ
SCHRAUBENDREHER PH2	ARBEITSBEREICH	PLATINE
SCHRAUBENDREHER SCHLITZ	WERKZEUGBEREICH	ABDECKUNG
HAMMER	UNDEFINED	KNAUF
UNDEFINED		UNDEFINED
Activity	Workflow	Task Complete
SCHRAUBEN	LOADWORKFLOW	DONE
POSITIONIEREN	STOPWORKFLOW	
AUSEINANDERNEHMEN		
UNDEFINED		

Abbildung 5.7: Benutzerschnittstelle zur Erfassung und Bereitstellung von Kontextinformationen

aller Kontextinformationen des Typs „Aktivität“ sowie der Kontextinformationen „Haube“ und „Platine“ des Typs „Material“. Für die Erkennung von Aktivitäten oder Arbeitsgesten stehen zum Zeitpunkt dieser Arbeit keine voll funktionsfähigen oder geeigneten Systeme zur Verfügung. Bzgl. der Kontextinformationen „Haube“ und „Platine“ ist eine händische Erfassung notwendig, da die Steuerplatine und die Haube des Stellantriebes aufgrund ihrer Größe separate Ablageorte besitzen. Eine automatisierte Erkennung ist somit nicht über die intelligente Materialbereitstellung möglich. Ferner liegen die Ablageorte außerhalb des Erfassungsbereiches der Tiefenbildkamera, sodass eine Erfassung durch die Objekterkennung ebenfalls nicht möglich ist.

**Vorbereitung der Nachrichtenkodierung**

Im bestehenden Assistenzsystem werden alle erfassten Informationen als Ereignisse betrachtet und es findet keine Einteilung in Informationskategorien oder -typen statt (vgl. Abschnitt 5.1). Damit eine Integration der entwickelten Verarbeitungskomponente mit den Systemkomponenten der Erfassung möglich ist, muss daher eine Anpassung der Nachrichtenstruktur erfolgen, um eine Aufteilung der Kontextinformationen entsprechend der definierten Kontexttypen (vgl. Abschnitt 4.3.3) zu erreichen. Hierzu wird eine Umwandlung der versendeten Nachrichten durchgeführt, indem alle versendeten Nachrichten ohne Zuordnung durch eine zentrale Stelle

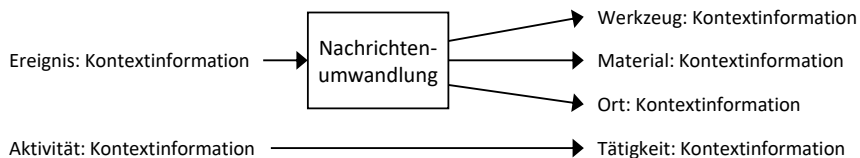


Abbildung 5.8: Schematische Darstellung der Nachrichtenumwandlung

empfangen und mit einer Zuordnung zu einem der Kontexttypen erneut versendet werden (vgl. Abbildung 5.8). Kontextinformationen, die bereits mit einer Zuordnung zum Kontexttyp versendet werden, werden nicht umgewandelt. Zu diesen gehören insbesondere die Nachrichten der GUI, der Objekterkennung und der intelligenten Materialbereitstellung, da deren Nachrichtenstruktur bereits bei der Umsetzung berücksichtigt wurde.

### Vorbereitung der aktuellen Verarbeitungskomponente (WVC)

Die Umwandlung des Nachrichtenformates stellt darüber hinaus auch den ersten Schritt zur Außerbetriebnahme der bestehenden Verarbeitungskomponente dar. Diese agiert auf der alten Nachrichtenstruktur ohne Zuordnung zu Kontexttypen und empfängt aufgrund der Nachrichtenumwandlung keine Ereignisse mehr, die zu einer Steuerung des Arbeitsprozesses führen. Damit eine einfache Anpassung der bestehenden Verarbeitungskomponente hin zum Workflow-Visualisierungs-Controller (WVC) erreicht werden kann, verwendet der WVC weiterhin das Workflow-Modell des bestehenden Assistenzsystems, um Arbeitsschritte zu identifizieren und die Visualisierung entsprechend dieser Arbeitsschritte (vgl. Abbildung 5.9) anzusteuern. Die neu entwickelte Verarbeitungskomponente und deren Teilkomponenten verwenden hingegen das *Workflow Model* entsprechend Abschnitt 4.3.5. Die Umsetzung des Modells erfolgt in Abschnitt 5.2.2.

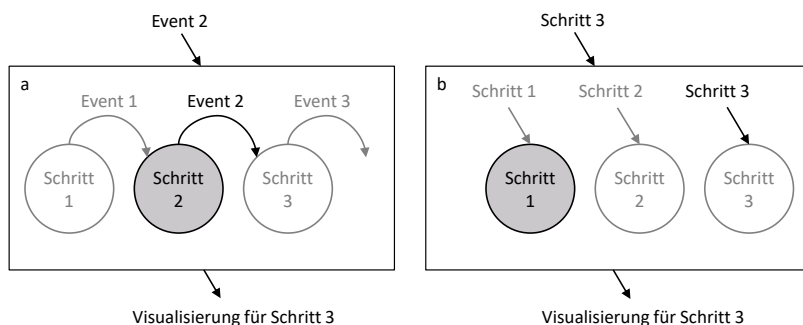


Abbildung 5.9: Verhalten des WVC vor der Anpassung (a) und nach der Anpassung (b) zur Integration der entwickelten Verarbeitungskomponente

Damit eine Verwendung des WVC zur Steuerung der Visualisierung auf Basis einer Interaktion mit der Verarbeitungskomponente möglich ist, wird zusätzlich die Verarbeitungslogik des WVC verändert. Statt auf eingehende Ereignisse zu warten und anhand dieser Ereignisse zu interpretieren, welcher Arbeitsschritt als nächstes ausgeführt wird (vgl. Abbildung 5.9a), wird die Verarbeitungslogik so verändert, dass der WVC auf Nachrichten der Verarbeitungskomponente bezüglich des erfassten Arbeitsschritts reagiert und in den entsprechenden Arbeitsschritt wechselt. Eine schematische Visualisierung des veränderten Verhaltens ist in Abbildung 5.9b dargestellt. Abbildung 5.9a zeigt das bisherige Verhalten, bei dem eine Erkennung des neuen Arbeitsschritts auf Basis eingehender Events und des aktuellen Arbeitsschritts durchgeführt wird. Das heißt ein Weiterschalteten erfolgt in einer festen Reihenfolge und nach vorherbestimmtem Verhalten. Bei der angepassten Verarbeitungslogik in Abbildung 5.9b reagiert das System hingegen direkt auf die Angabe des gemeldeten Arbeitsschrittes und ist unabhängig vom aktuell ausgewählten Arbeitsschritt. Die in Abbildung 5.9 grau hinterlegten Schritte zeigen jeweils einen aktuell ausgewählten Arbeitsschritt. Die schwarzen Übergangspfeile zeigen in welchen Arbeitsschritt als Reaktion auf die eingehende Nachricht gewechselt wird.

Die praktische Umsetzung dieses veränderten Verhaltens erfolgt durch eine Anpassung der Informationen im bestehenden Workflow-Modell. Hierzu werden die Ereignisse, die die Arbeitsschritte verketten, abgeändert. Einerseits werden weitere Ereignisse hinzugefügt, sodass jeder Arbeitsschritt mit jedem anderen Arbeitsschritt verknüpft ist, andererseits werden die spezifischen Ereignisnamen so verändert, dass der Name jedes Ereignisses dem Namen des Zielarbeitsschritts entspricht (vgl. Abbildung 5.10). Durch dieses Vorgehen kann der WVC von jedem Arbeitsschritt in jeden anderen Arbeitsschritt wechseln und tut dies immer dann, wenn der Name eines Arbeitsschritts übermittelt wird.

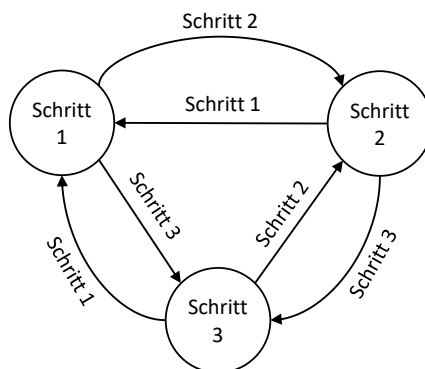


Abbildung 5.10: Schematische Darstellung der veränderten Verknüpfung von Arbeitsschritten des bestehenden Workflow-Modells



### Versuchsaufbau für die Umsetzung des Anwendungsszenarios

Abbildung 5.11 zeigt den resultierenden physischen Aufbau des Handarbeitsplatzes nach der Erweiterung der Komponenten im Bereich der Erfassung und der Anpassung am bestehenden Assistenzsystem. Physische Veränderungen im Vergleich zum ursprünglichen Aufbau (vgl. Abbildung 5.4 in Abschnitt 5.1) zeigen sich vor allem im Bereich der Ablageboxen mit den ergänzten Wägezellen (umrahmter Bereich). Alle weiteren Veränderungen beziehen sich vornehmlich auf die Ergänzung und Anpassung von Softwarekomponenten und sind somit nicht direkt ersichtlich.

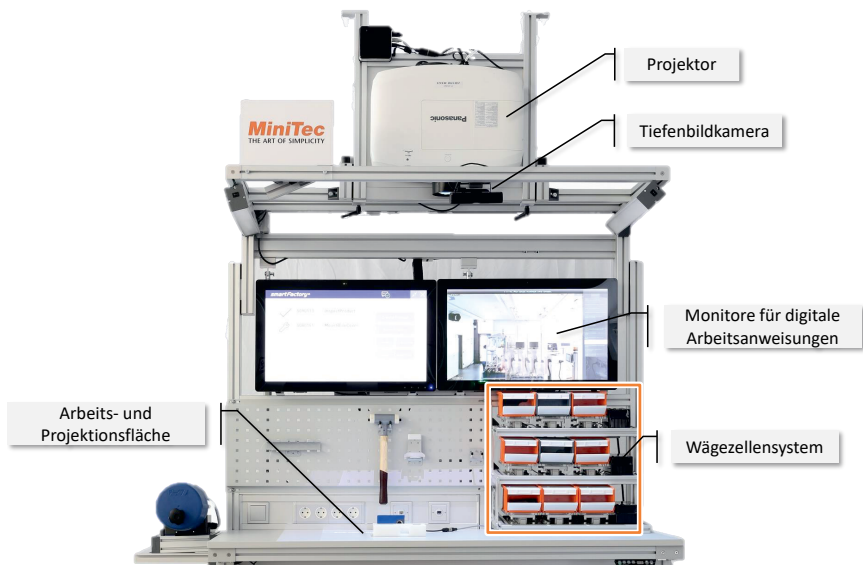


Abbildung 5.11: Aufbau und Bestandteile des Handarbeitsplatzes der SmartFactory<sup>KL</sup>

Der Versuchsaufbau für die Evaluation und Umsetzung der Verarbeitungskomponente wird in Abbildung 5.12 verdeutlicht. Die Durchführung von Montage- und Demontageschritten des Arbeitsprozesses erfolgt dabei im dargestellten Arbeitsbereich. Für die Kontexterfassung der Tiefenbildkamera ist dieser Bereich als Arbeitsbereich definiert. Die Werkzeuge, die für den Arbeitsprozess benötigt werden, hängen im gekennzeichneten Bereich hinter dem Arbeitsraum. Auch hier gilt, dass der Werkzeugbereich durch die Tiefenbildkamera als solches erfasst wird. Als Ablagebereiche für die demontierten Bauteile werden die mit Box 1 bis Box 4 bezeichneten Materialcontainer der intelligenten Materialbereitstellung verwendet. Zusätzlich stehen die markierten Ablageplätze links des Arbeitsbereiches bereit, um die Steuerplatine und die Haube abzulegen. Für das Anwendungsszenario wird die folgende Zuordnung der Materialien festgelegt: „Knauf“ in Box 1, „Schrauben M3x10“ in Box 2 und „Abdeckung“ in Box 4.

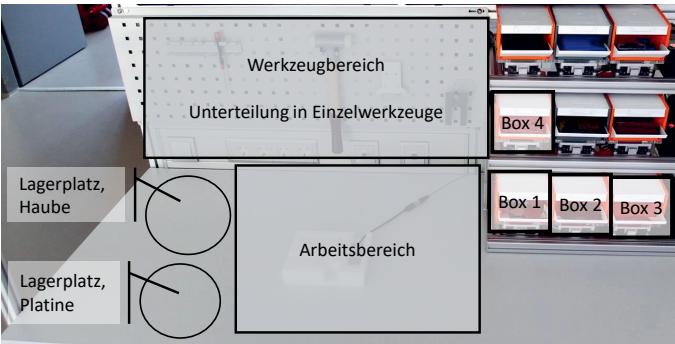


Abbildung 5.12: Versuchsaufbau des bestehenden Systems für die Umsetzung des Anwendungsszenarios

### 5.2.2 Umsetzung des Workflow-Modells

Bevor eine Umsetzung der Verarbeitungskomponente und des Anwendungsszenarios durchgeführt werden kann, muss der Arbeitsprozess (siehe Tabelle 5.1 in Abschnitt 5.1) entsprechend der Arbeitsprozessmodellierung in Abschnitt 4.3.2 vorbereitet werden. Hierzu werden relevante Kontextinformationen für die einzelnen Arbeitsschritte ergänzt, die auf Basis der Aufgabenbeschreibung des Arbeitsschritts sowie der verwendeten Materialien und Werkzeuge hergeleitet werden. Um eine semantische Beschreibung und automatisierte Identifikation zu erreichen, werden die Material- und Werkzeugbezeichnungen aus bekannten Klassifikationssystemen (vgl. Abschnitt 2.2.4) entnommen. Die Bezeichnungen der Aktivitäten ergeben sich aus eigenen Vorarbeiten zusammen mit Wansch (vgl. Abschnitt 2.2.3). Die Ergebnisse dieser Ergänzung und Anpassung sind in Tabelle 5.2 zusammengefasst und werden im weiteren Verlauf für die Umsetzung des Workflow-Modells verwendet. Eine Verkettung der Arbeitsschritte durch die kausalen Abhängigkeiten (Vorbedingungen) bleibt entsprechend Tabelle 5.1 weiterhin erhalten, ist aus Gründen der Übersichtlichkeit jedoch nicht in der Tabelle aufgeführt.

Tabelle 5.2: Informationen für die Modellierung des Arbeitsprozesses des Anwendungsszenarios

ID	Aufgabe	Werkzeug	Aktivität	Material	Ort
0	Knauf lösen	Innensechskant 4mm	Schrauben	-	Werkzeugbereich Arbeitsbereich
1	Knauf ablegen	-	Auseinandernehmen	Knauf	Arbeitsbereich Materiallager
2	Haube ablegen	-	Auseinandernehmen	Haube	Arbeitsbereich

ID	Aufgabe	Werkzeug	Aktivität	Material	Ort
3	Abdeckung lösen	Schraubendreher PH1	Schrauben	-	Werkzeugbereich Arbeitsbereich
4	Abdeckung ablegen	-	Auseinandernehmen	Abdeckung	Arbeitsbereich Materiallager
5	Litzen lösen	Schraubendreher Schlitz	Schrauben	-	Werkzeugbereich Arbeitsbereich
6	Platine lösen	Schraubendreher PH2	Schrauben	-	Werkzeugbereich Arbeitsbereich
7	Schrauben ablegen	-	Auseinandernehmen	Schrauben M3x10	Arbeitsbereich Materiallager
8	Platine ablegen	-	Auseinandernehmen	Platine	Arbeitsbereich
9	Neue Platine einsetzen	-	Positionieren	Platine	Arbeitsbereich
10	Schrauben einsetzen	-	Positionieren	Schrauben M3x10	Materiallager Arbeitsbereich
11	Platine befestigen	Schraubendreher PH2	Schrauben	-	Werkzeugbereich Arbeitsbereich
12	Litzen befestigen	Schraubendreher Schlitz	Schrauben	-	Werkzeugbereich Arbeitsbereich
13	Abdeckung einsetzen	-	Positionieren	Abdeckung	Materiallager Arbeitsbereich
14	Abdeckung befestigen	Schraubendreher PH1	Schrauben	-	Werkzeugbereich Arbeitsbereich
15	Haube befestigen	-	Positionieren	Haube	Arbeitsbereich
16	Knauf einsetzen	-	Positionieren	Knauf	Materiallager Arbeitsbereich
17	Knauf befestigen	Innensechskant 4mm	Schrauben	-	Werkzeugbereich Arbeitsbereich

Die Umsetzung des Workflow-Modells erfolgt auf Basis der Informationen zum Arbeitsprozess sowie der entwickelten Struktur des Workflow-Modells (vgl. Abschnitt 4.3.5). Um die Daten für das Workflow-Modell bereitzustellen, wird im ersten Schritt eine Datenbank angelegt, die alle Informationen zum Arbeitsprozess entsprechend Tabelle 5.1 und Tabelle 5.2 enthält. Die Struktur der Datenbank orientiert sich an der Darstellung der Informationen in den Tabellen und an der Modellierung der Verarbeitungslogik in Abschnitt 4.3.4. Während die Arbeitsschritte in Tabelle 5.1 und Tabelle 5.2 jedoch durch eine eindeutige numerische ID identifiziert werden,

	id	name	tool	activity	material	place	predecessorid1	predecessorid2
▶	1	LoosenStarKnobComplete	Innensechskant	Schrauben	undefined	Arbeitsbereich	NULL	NULL
	2	UnscrewCoverComplete	PH 1	Schrauben	undefined	Arbeitsbereich	NULL	NULL
	3	RemoveStarKnobComplete	undefined	Auseinandernehmen	Knauf	Arbeitsbereich	1	NULL
	4	RemoveCoverComplete	undefined	Auseinandernehmen	Abdeckung	Arbeitsbereich	2	NULL
	5	RemoveHoodComplete	undefined	Auseinandernehmen	Haube	Arbeitsbereich	3	NULL

Abbildung 5.13: Ausschnitt aus der Datenbank zur Repräsentation des Workflow-Modells

baut das entwickelte System auf einer Identifikation mittels eindeutiger Zeichenkette auf, um eine semantische Interpretation der Daten zu gewährleisten. Die Festlegung der Zeichenkette basiert auf einer prägnanten Bezeichnung des Arbeitsschritts in englischer Sprache. Zur einheitlichen Notation wird die *Pascal Case* oder *Upper Camel Case* Schreibweise<sup>15</sup> verwendet. Abbildung 5.13 zeigt einen Ausschnitt aus der realisierten Datenbank. Bezogen auf die Systemarchitektur in Abschnitt 4.2 entspricht die Datenbank dem Workflow-Speicher, über den die Workflow-Modelle für die Verarbeitungskomponente bereitgestellt werden. Auf die verschiedenen Möglichkeiten zur Umsetzung einer Datenbank als Workflow-Speicher wird im Rahmen dieser Arbeit nicht eingegangen. Dies gilt ebenso für eine einfache oder automatisierte Erstellung von Arbeitsprozessen zur Befüllung der Datenbank. Jedoch existieren bereits mehrere Arbeiten, die sich insbesondere mit dem Bereich der automatisierten Herleitung von Arbeitsschritten und Montagereihenfolgen aus CAD-Dateien beschäftigen (vgl. [BB16; Ha18; Zh17]).

Die Verwendung des Workflows durch die Verarbeitungskomponente erfolgt über die Interpretation der Datenbankinformationen mithilfe des Workflow-Modells entsprechend Abschnitt 4.3.5. Der zweite Schritt der Umsetzung ist daher die Realisierung des UML-Klassendiagramms dieses Workflow-Modells. Abbildung 5.14 zeigt einen Ausschnitt des mit Informationen befüllten Workflow-Modells zu einem spezifischen Zeitpunkt im Verlaufe des Arbeitsprozesses. Zu dem gewählten Zeitpunkt ist eine Bearbeitung der Arbeitsschritte „Abdeckung lösen“ und „Knauf abnehmen und ablegen“ möglich. Der Arbeitsschritt „Knauf nehmen und ablegen“ gilt dabei als in Bearbeitung, sodass eine Unterstützung für diesen Schritt bereitgestellt wird. Die Annotation der Arbeitsschritte mit ihrer jeweiligen optimalen Beobachtungen (vgl. Abschnitt 4.3.3) wird durch eine Verknüpfung mit Context-Objekten realisiert. Das Task-Objekt ohne ID und ohne Beschreibung steht repräsentativ für alle nicht dargestellten Arbeitsschritte. Die softwaretechnische Implementierung wird mittels der Programmiersprache C# durchgeführt.

<sup>15</sup>vgl. [Gr01]. Unter der *Pascal Case* oder *Upper Camel Case* Schreibweise werden gebräuchliche Binnenversalien in Programmiersprachen verstanden, die zur besseren Übersichtlichkeit verwendet werden.

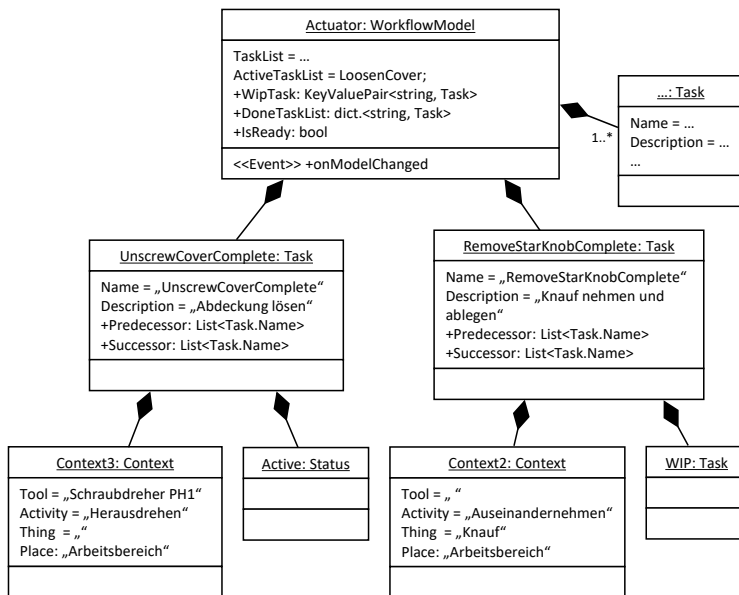


Abbildung 5.14: Ausschnitt des Workflow-Modells am Beispiel des Anwendungsszenarios in Form von UML-Objekten

### 5.2.3 Umsetzung der Verarbeitungskomponente

Um die Umsetzung und Implementierung der Verarbeitungslogik durchzuführen, erfolgt zuerst die Ausspezifizierung des Messaging Client aus Abschnitt 4.3.5, der als Verbindungsglied zwischen der Verarbeitungskomponente und dem Assistenzsystem dient. Zur Integration der Verarbeitungskomponente in das *SmartFactory*<sup>KL</sup>-Assistenzsystem wird eine Anbindung an den MQTT Kommunikations-Broker durchgeführt.

Entsprechend dieses Kommunikationsprotokolls wird die Verarbeitungslogik als eigenständige Softwarekomponente umgesetzt, die sich als Teilnehmer im MQTT-Verbund anmeldet und dadurch in das Assistenzsystem integriert wird. Aufgrund der modellbasierten Entwicklung der Verarbeitungskomponente ist diese jedoch nicht auf eine Umsetzung mit dem MQTT-basiertem Messaging Client beschränkt. Stattdessen kann eine Umsetzung für andere Kommunikationsprotokolle wie OPC UA erfolgen. Ferner kann eine Umsetzung mit universellem Messaging Client, der verschiedene Kommunikationsprotokolle beherrscht, durchgeführt werden. Aufgrund der Zielsetzung und Abgrenzung dieser Arbeit liegt die Implementierung eines solchen Systems nicht im Fokus und wird nicht weiter betrachtet.

Tabelle 5.3: Übersicht der verwendeten MQTT-Topics

MQTT Nachrichten	Inhalt
HAP/Context/Tool	Information über das aktuell verwendete Werkzeug
HAP/Context/Activity	Information über die aktuell durchgeführte Aktivität
HAP/Context/Material	Information über das aktuell verwendete Bauteil / Material
HAP/Context/Place	Information über den Ort / Bereich, indem aktuell etwas getan wird
HAP/Context/Request	Anfrage, welche Kontextinformationstypen bereitgestellt werden
HAP/Context/Type	Information über die aktuell bereitgestellten Kontextinformationstypen (Tool, Activity, Material, Place)
HAP/Control/Task/Complete	Steuerinformation darüber, dass ein Arbeitsschritt erfolgreich bearbeitet wurde
HAP/Control/Task/Show	Steuerinformationen, um einen Arbeitsschritt explizit anzuzeigen
HAP/Control/Workflow/Load	Steuerinformation, um den angegebenen Workflow zu laden
HAP/Control/Workflow/Stop	Steuerinformation, um den aktuellen Workflow zu stoppen
HAP/Workflow/Task	Information zum Arbeitsschritt für den eine Unterstützung angezeigt werden soll
HAP/Workflow/Finished	Information zum erfolgreichen Abschluss des Arbeitsprozesses

Die Nachrichtentopics, über die eine Interaktion und Kommunikation mit der Verarbeitungskomponente durchgeführt werden, richten sich nach den Aufgaben, die die Verarbeitungskomponente zu erledigen hat. Für die Teilkomponente der *Task Recognition* ist das die Erfassung des aktuell durchgeführten Arbeitsschritts, anhand übermittelter Kontextinformationen. Für die Übermittlung der Kontextinformationen werden hierzu die Topics „HAP/Context/#“ verwendet. Das #-Symbol dient dabei als Platzhalter für die vier in Abschnitt 4.3.3 vorgestellten Kontexttypen (vgl. Tabelle 5.3). Die Überprüfung durch den *Context Type Matcher*, ob alle Kontexttypen durch das Assistenzsystem bereitgestellt werden, erfolgt ebenfalls unter diesem Topic. Hierzu versendet der *Context Type Matcher* eine Anfrage an das System (HAP/Context/Request) und empfängt Nachrichten zu den bereitgestellten Typen (HAP/Context/Type). Die Teilkomponente des *Workflow Managers* kommuniziert über die Topics „HAP/Control/#“ für den Empfang von Informationen zur Steuerung der Verarbeitungskomponente und „HAP/Workflow/#“ für Benachrichtigungen zum Arbeitsprozess und dem aktuellen Arbeitsschritt. Eine Übersicht über alle Nachrichtentopics ist in Tabelle 5.3 aufgeführt.

Nachdem die Spezifizierung des Messaging Clients durchgeführt wurde und die zur Interaktion notwendigen Nachrichtentopics definiert wurden, erfolgt im nächsten Schritt die Umsetzung der Teilkomponente des *Workflow Managers*. Die Hauptaufgaben des *Workflow Managers* sind

die Aktualisierung und die Verwaltung des Workflow-Modells sowie die Übermittlung des aktuellen Arbeitsschritts an die Visualisierung (vgl. Abschnitt 4.3.5 und Abbildung 4.13).

Zur ersten Hauptaufgabe gehört das Laden eines neuen Workflows aus der Datenbank und die Übertragung der Arbeitsschritte in das Workflow-Modell. Zusätzlich wird das Workflow-Modell initialisiert, indem die Arbeitsschritte über ihre jeweiligen Vorbedingungen miteinander verknüpft werden und der Status aller Arbeitsschritte aktualisiert wird (vgl. Codeausschnitt 5.1).

---

```
if (topic == MqttMessaging.LoadWorkflow)
{
    // Workflow laden und die Liste der Arbeitsschritte übergeben
    // Abspeichern der Schritte im Workflow-Modell
    WorkflowModel.tasklist = WorkflowController.loadWorkflow(message);

    // Initialer Status und Vorbedingungen der Arbeitsschritte festlegen
    WorkflowController.initializeTaskStatus();

    // Liste mit den erledigten Arbeitsschritten aktualisieren und
    // aktive Arbeitsschritte bestimmen
    WorkflowController.updateTaskStatus();
}
```

---

Codeausschnitt 5.1: Initialisierung eines neuen Workflows

Im laufenden Betrieb werden die Arbeitsschrittstatus nach jedem erfolgreichen Abschluss eines Arbeitsschritts aktualisiert. Bei dieser Aktualisierung wird das gesamte Workflow-Modell auf Basis der aktuellen Status jedes Arbeitsschritts geprüft und neu generiert. Codeausschnitt 5.2 zeigt den Prozess der Aktualisierung, bei dem zuerst die Liste an erledigten Arbeitsschritten angelegt wird. Nach einer Bestimmung der aktiven Arbeitsschritte wird zudem die Liste der aktiven Arbeitsschritte neu erstellt. Die Unterteilung der gesamten Liste an Arbeitsschritten in Teillisten dient dem einfacheren Zugriff auf die relevanten Arbeitsschritte und der besseren Repräsentation des Arbeitsprozesszustandes.

---

```
public static void updateTaskStatus()
{
    // Liste mit den erledigten Arbeitsschritten neu anlegen
    setDoneTasklist();

    // Alle Arbeitsschritte auf Active setzen, deren Vorbedingungen erfüllt
    // sind
    // und die noch nicht erledigt sind
    checkActiveTasks();

    // Liste mit aktiven Arbeitsschritten neu anlegen
    setActiveTasklist();
}
```

---

Codeausschnitt 5.2: Aktualisierung der Arbeitsschrittstatus im Workflow-Modell

Der erfolgreiche Abschluss eines Arbeitsschritts wird entweder durch die *Task Recognition* erkannt oder kann durch explizite Bestätigung über die Benutzerschnittstelle erfolgen (vgl. Abschnitt 5.2.1). In beiden Fällen wird der Arbeitsschritt, der sich aktuell in Bearbeitung befindet, als erledigt markiert. Hierzu wird das Workflow-Modell aktualisiert (vgl. Codeausschnitt 5.3) und von der *Task Recognition* ein neuer Arbeitsschritt als „in Bearbeitung“ festgelegt.

---

```

public static void completeWipTask()
{
    if (WorkflowModel.wipTask != null)
    {
        // In Bearbeitung befindliche Arbeitsschritte als erledigt markieren
        WorkflowModel.wipTask.status = "done";
        WorkflowModel.wipTask = null;

        // Aktualisierung des Workflow-Modells
        updateTaskStatus();
    }
}

```

---

Codeausschnitt 5.3: Markierung des sich aktuell in Bearbeitung befindlichen Arbeitsschritts als „erledigt“

Der Wechsel oder die Festlegung eines Arbeitsschritts als „in Bearbeitung“ erfolgt sobald die *Task Recognition* diesen anhand der Kontextinformationen zur aktuellen Arbeitssituation erkannt hat. Zwar werden durch die *Task Recognition* nur Arbeitsschritte erkannt, die zum entsprechenden Zeitpunkt durchführbar sind, jedoch überprüft der *Workflow Manager* die Validität des gemeldeten Arbeitsschritts, bevor dieser als „in Bearbeitung“ markiert wird (vgl. Codeausschnitt 5.4). Ein durchgeführter Wechsel des Arbeitsschritts wird anschließend über den Messaging Client an das Assistenzsystem übermittelt.

---

```

public static void changeWipTask(int id_wip)
{
    // Prüfung ob ein Wechsel des Arbeitsschrittes stattfindet
    if (WorkflowModel.tasklist[id_wip].status == "Active")
    {
        // Wechsel des Arbeitsschritts der aktuell als in Bearbeitung gilt
        // mit dem neu erkannten Arbeitsschritt
        if (WorkflowModel.wipTask != null)
            WorkflowModel.wipTask.status = "Active";
        WorkflowModel.wipTask = WorkflowModel.tasklist[id_wip];
        WorkflowModel.wipTask.status = "WIP";

        // Benachrichtigung an das Assistenzsystem über den Arbeitsschritt
        // der als
        // in Bearbeitung erkannt wurde
        MqttMessaging.publish("HAP/Workflow/Task",
            WorkflowModel.wipTask.name);
    }
}

```

---

Codeausschnitt 5.4: Wechsel des Arbeitsschritts der als „in Bearbeitung“ erkannt wurde

Im dritten Schritt der Umsetzungsphase wird die Teilkomponente der *Task Recognition* implementiert, deren Hauptaufgabe die Erkennung von Arbeitsschritten auf Basis von empfangenen Kontextinformationen zur aktuellen Arbeitssituation darstellt. Im Gegensatz zur Teilkomponente des *Workflow Managers* existiert für die *Task Recognition* keine eindeutige Arbeitsweise, nach der eine Umsetzung erfolgen kann. Die Arbeitsweise der Teilkomponenten wird maßgeblich durch die Menge der bereitstehenden Kontextinformationen, die Kontextinformationstypen sowie die Bewertungsfunktion zur Ermittlung des Übereinstimmungsgrades bestimmt (vgl. Abschnitt 4.3.3). Da eine konkrete Definition der geeignetsten Bewertungsfunktion und der dafür notwendigen Kontextinformationen ohne gezielte Simulationen oder Expe-



rimente nicht möglich ist, wird die *Task Recognition* für zwei Arbeitsweisen umgesetzt. Das Grundprinzip der Arbeitsweise beruht dabei für beide Varianten auf einem eins-zu-eins Vergleich der erwarteten Kontextinformationen (optimale Beobachtung) und den übermittelten Kontextinformationen. Die Berechnung der Übereinstimmung ergibt sich daher aus der Summe der Einzelübereinstimmungen, wobei jede Einzelübereinstimmung gleich gewertet wird. Das heißt, bei einem Vergleich von vier Kontextinformationen entspricht jede Einzelübereinstimmung einem Anteil von  $\frac{1}{4}$ . Bei einem Vergleich von sechs Kontextinformationen entspricht jede Einzelübereinstimmung einem Anteil von  $\frac{1}{6}$ .

Entsprechend dieses Grundprinzips unterscheiden sich die zwei Arbeitsweisen durch die Anzahl der betrachteten Kontextinformationen. Arbeitsweise eins führt eine Berechnung der Übereinstimmung auf Basis von vier Kontextinformationen durch, wobei jeweils eine Kontextinformation je Kontexttyp berücksichtigt wird. Bei Arbeitsweise zwei werden sechs Kontextinformationstypen betrachtet, wobei für die Kontexttypen „Aktivität“, „Werkzeug“ und „Material“ jeweils eine Kontextinformationen betrachtet wird. Bei Kontextinformationen des Typs „Ort“ hingegen werden die drei zuletzt gemeldeten Kontextinformationen für die Berechnung der Übereinstimmung gewertet. Um eine Umsetzung dieser Arbeitsweise zu ermöglichen, wird zusätzlich eine Änderung am Workflow-Modell durchgeführt, sodass eine Verknüpfung von jeweils drei Kontextinformationen des Typs „Ort“ möglich ist. Hintergrund dieser Arbeitsweise ist die Abbildung von Kontextinformationssequenzen, um eine bessere Beschreibung der Arbeitssituation zu erreichen. Wird in einem Arbeitsschritt beispielsweise ein Werkzeug benötigt, so muss die nutzende Person zuerst das Werkzeug aus dem Werkzeugbereich holen, bevor diese das Werkzeug im Arbeitsbereich verwenden und anschließend wieder im Werkzeugbereich deponieren kann.

Zusätzlich zur kontextsensitiven Arbeitsschritterkennung übernimmt die *Task Recognition* die Aufgabe der Bestimmung, ab wann ein Arbeitsschritt erfolgreich bearbeitet wurde. Ähnlich zur Arbeitsschritterkennung kann ohne Experimente und Nutzerstudien keine eindeutige Arbeitsweise definiert werden, mit der eine optimale Bestimmung eines erfolgreich bearbeiteten Arbeitsschritts möglich ist. Für die finale Umsetzung der Verarbeitungskomponente wird jedoch die Annahme getroffen, dass ein Arbeitsschritt bei einer vollständigen Übereinstimmung als erfolgreich bearbeitet angesehen werden kann.

Auf eine Umsetzung des *Context Type Matchers* wurde im Rahmen dieser Arbeit verzichtet, da für die Evaluation sichergestellt wird, dass Informationen zu allen Kontexttypen übermittelt werden. Ferner leistet der *Context Type Matcher* keinen Beitrag zur kontextsensitiven Bestimmung des Arbeitsschritts, der gerade bearbeitet wird, oder zur Verwaltung des Workflow-Modells. Aus diesen Gründen würde eine Umsetzung und Integration des *Context Type Matchers* zu keiner Veränderung der Evaluation oder der Testergebnisse führen.

### 5.3 Komponententest

Das Ziel des Komponententests ist die Überprüfung der Funktionsweise der Verarbeitungslogik. Der Test wird losgelöst vom gesamten Assistenzsystem durchgeführt, um Fehler in der modellierten Arbeitsweise der Verarbeitungskomponente festzustellen. Ferner erfolgt eine Evaluation der Qualität der Arbeitsschitterkennung, um Probleme bzw. Verbesserungspotenziale für die Verarbeitungslogik abzuleiten. Durch die Durchführung des Tests, ohne eine Integration in das Gesamtsystem, werden Störeinflüsse anderer Komponenten vermieden und eine Auswirkung auf die Evaluation der Verarbeitungskomponente ausgeschlossen. Damit ein Test der vollständigen Funktionsweise der Verarbeitungslogik möglich ist, wird ein Testprogramm umgesetzt, das die notwendigen Interaktionen mit anderen Komponenten des Assistenzsystems simuliert und über MQTT mit der Verarbeitungskomponente kommuniziert. Entsprechend dem gewählten Anwendungsszenario (vgl. Abschnitt 5.1) übernimmt das Testprogramm folgende Aufgaben:

- Start des Workflows „Reparatur des Stellantriebes“
- Bereitstellung von Kontextinformationen für alle Kontexttypen
- Erfassung der Arbeitsschritte, die an den WVC gemeldet werden
- Erstellung von Logdateien zur Protokollierung des Testverlaufs

Durch dieses Vorgehen wird für jeden gestarteten Testlauf ein Ereignis-Log angelegt, der von Start des Workflows bis zur Beendigung des Workflows sequenziell aufzeichnet, welche Kontextinformationen der Verarbeitungskomponente übermittelt wurden und welche Arbeitsschritte als Reaktion erkannt und an die Visualisierung gemeldet wurden. Die Kontextinformationen, die das Testprogramm bereitstellt, werden durch einen Zufallsalgorithmus bestimmt. Dieser wählt für jede Interaktion eine Kontextinformation aus, die an die Verarbeitungskomponente übermittelt wird. Hierbei wird zuerst ein zufälliger Kontexttyp ausgewählt. Anschließend wählt der Algorithmus aus allen verfügbaren Kontextinformationen dieses Typs eine Kontextinformation aus. Als verfügbar gelten dabei alle Kontextinformationen, die im Workflow-Modell mit mindestens einem Arbeitsschritt verknüpft sind.

Der sequenzielle Ablauf eines Testlaufs und die Interaktion zwischen Testprogramm und Verarbeitungskomponente während dieses Testlaufs sind exemplarisch in Abbildung 5.15 als Sequenzdiagramm dargestellt. Die Beschriftung der Interaktionen repräsentiert die übermittelte MQTT-Nachricht, bestehend aus MQTT-Topic gefolgt vom Inhalt der Nachricht in Klammern. Kursiv geschriebene Inhalte oder kursive Teile des MQTT-Topics repräsentieren Platzhalter. Während eines Testlaufs werden diese Platzhalter durch konkrete Informationen ersetzt. Weiterhin zeigt die Abbildung keine fest definierte Reihenfolge der Interaktionen und MQTT-Nachrichten, da sich die Reihenfolge der Interaktionen abhängig vom tatsächlichen Ablauf und den übermittelten Nachrichten ergibt. Der Beginn und das Ende eines Testlaufs, indem der Workflow geladen und beendet sowie die Log-Dateien ausgegeben wird, sind jedoch bei allen Testläufen gleich.

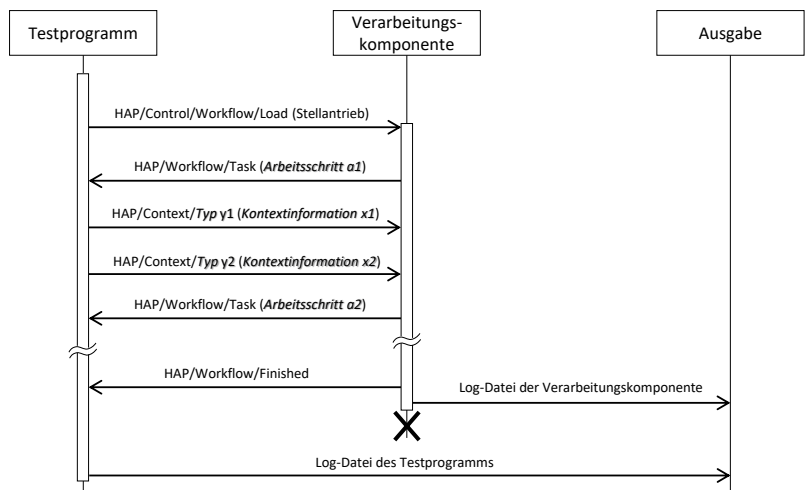


Abbildung 5.15: Sequenzdiagramm zu den Interaktionen zwischen Testprogramm und Verarbeitungskomponente

Die Ereignis-Logs, die für jeden Testlauf erstellt werden, dienen der Überprüfung der in Tabelle 5.4 aufgeführten Kriterien, um die Funktionsweise der Verarbeitungskomponente zu evaluieren. Das Log des Testprogramms erfassen die über MQTT ausgetauschten Informationen, das heißt die bereitgestellten Kontextinformationen sowie eine Meldung über Arbeitsschritte, die auf Basis dieser Kontextinformationen erkannt wurde. Aus diesem Grund können mithilfe des Logs die Kriterien K1 bis K4 bewertet werden.

Tabelle 5.4: Kriterien zur Evaluation der Verarbeitungskomponente

ID	Kriterium
K1	Start des Workflows durch das System
K2	Beendigung des Workflows nach Bearbeitung durch die Verarbeitungskomponente
K3	Reaktion der Verarbeitungskomponente auf Kontextinformationen
K4	Information über Arbeitsschritte, die durch die Verarbeitungskomponente erfasst wurden
K5	Kausal richtige Reihenfolge der Arbeitsschritte
K6	Bestimmung der zu aktuellem Zeitpunkt durchführbaren Arbeitsschritte
K7	Auswahl des wahrscheinlichsten Arbeitsschritts entsprechend der Kontextinformationen

Für eine Bewertung der Kriterien K5 bis K7 wird zusätzlich eine Log-Datei der Verarbeitungskomponente erstellt. Diese speichert die internen Entscheidungen der Verarbeitungskomponente sowie den Zustand des Workflow-Modells zu den jeweiligen Zeitpunkten. In Anlehnung an die Log-Datei des Testprogramms folgt der Aufbau dieser Log-Datei dem sequenziellen Ablauf eines Testlaufs. Dementsprechend wird für jede eingehende MQTT-Nachricht ein neuer Eintrag angelegt. Der Inhalt der Logeinträge gliedert sich dabei in jeweils einen Bereich zur Repräsentation

- der zuletzt eingegangenen Kontextinformation je Kontexttyp,
- der Arbeitsschritte, die zu diesem Zeitpunkt durchgeführt werden können,
- des Grads der Übereinstimmung für jeden durchführbaren Arbeitsschritt und
- des Arbeitsschritts, der durch die Verarbeitung als „in Bearbeitung“ gewählt wurde.

Abbildung 5.16 zeigt beispielhaft den Ausschnitt aus einer Log-Datei der Verarbeitungskomponente. Zur Verbesserung der Übersichtlichkeit sind nicht alle Iterationen dargestellt, da diese als Duplikate zu den dargestellten Iterationen gewertet werden können (Vergleich Iteration 111 und 112). Ausgeblendete Iterationen sind durch die Trennung mit einer Doppellinie kenntlich gemacht.

Die entsprechende Log-Datei des Testprogramms zu dem in Abbildung 5.16 gezeigten Ausschnitt ist in Abbildung 5.17 dargestellt. Ausgeblendete Zeilen, um die Übersichtlichkeit zu verbessern, sind ebenfalls mit einer Trennung durch eine Doppellinie kenntlich gemacht. Bei einem Vergleich zwischen Abbildung 5.16 und Abbildung 5.17 zeigt sich, dass vom Testprogramm bereitgestellte Kontextinformationen durch die Verarbeitungskomponente erfasst und verarbeitet werden. Weiterhin ist zu sehen, dass die Verarbeitungskomponente einen Wechsel des Arbeitsschritts durchführt, der als „in Bearbeitung“ erkannt wird. Die beiden Log-Dateien bieten damit eine Repräsentation desselben Testlaufs aus zwei unterschiedlichen Blickwinkeln.

Iteration	Task Name	Place	Tool	Activity	Material	Status	%
..	..	..	..	..	..	..	..
111	LoosenPCBScrewsComplete	Arbeitsbereich		Schrauben		wip	75
	LoosenScrewsComplete					active	75
112	LoosenPCBScrewsComplete	Arbeitsbereich		Schrauben		wip	75
	LoosenScrewsComplete					active	75
116	LoosenPCBScrewsComplete	Arbeitsbereich			Schrauben, Innensechskant	wip	25
	LoosenScrewsComplete					active	25
117	LoosenPCBScrewsComplete	Arbeitsbereich			Platine	wip	25
	LoosenScrewsComplete					active	25
119	LoosenPCBScrewsComplete	Arbeitsbereich		Schrauben		wip	75
	LoosenScrewsComplete					active	75
121	LoosenPCBScrewsComplete	Arbeitsbereich	PH 1	Schrauben		wip	100
	LoosenScrewsComplete					active	75
122	LoosenScrewsComplete	Arbeitsbereich	PH 1	Schrauben		wip	75
	RemoveScrews					active	25
..	..	..	..	..	..	..	..

Abbildung 5.16: Ausschnitt aus einer Log-Datei der Verarbeitungskomponente

Die Durchführung des Komponententests erfolgt mittels drei Testszenarien, in denen jeweils vier Testläufe absolviert werden. Anschließend werden die Ergebnisse ausgewertet und für jedes Testszenario zusammengefasst. Tabelle 5.5 bis Tabelle 5.7 präsentieren diese Zusammenfassungen der Testläufe je Szenario.

Nr.	Ereignis	Inhalt
1	HAP/Control/Workflow/Load	Stellantrieb
102	HAP/Workflow/Task	LoosenPCBScrewsComplete
119	HAP/Context/Activity	Schrauben
120	HAP/Context/Place	Arbeitsbereich
123	HAP/Context/Activity	
124	HAP/Context/Material	Schrauben, Innensechskant
125	HAP/Context/Material	Platine
126	HAP/Context/Material	
127	HAP/Context/Aktivität	Schrauben
129	HAP/Context/Tool	PH1
130	HAP/Workflow/Task	LoosenScrewsComplete
...	HAP/Workflow/Finished	

Abbildung 5.17: Ausschnitt einer Log-Datei des Testprogramms

#### Testszenario 1:

Nach dem Start des Workflows werden Kontextinformationen in einem zeitlichen Abstand von 10 Millisekunden bereitgestellt. Die Auswahl der Kontextinformationen erfolgt entsprechend der Beschreibung zum Testprogramm in diesem Kapitel. Die erfolgreiche Bearbeitung eines Arbeitsschritts wird auf Basis der Übereinstimmungswahrscheinlichkeit der Verarbeitungskomponente durchgeführt. Erreicht ein Arbeitsschritt eine Übereinstimmungswahrscheinlichkeit von 100 % wird dieser als bearbeitet angesehen und entsprechend der Logik in Abschnitt 4.3.5 markiert. Die Verarbeitung arbeitet auf Basis von vier Kontextinformationen.

#### Testszenario 2:

Das zweite Testszenario unterscheidet sich vom ersten Szenario in der Art und Weise wie ein Arbeitsschritt als erfolgreich bearbeitet erkannt wird. Im Gegensatz zu einer Erkennung bei einer Übereinstimmungswahrscheinlichkeit von 100 % erfolgt eine Erkennung bei einer Übereinstimmungswahrscheinlichkeit von  $\geq 75$  %.

**Testszenario 3:** Für das dritte Testszenario gelten die gleichen Rahmenbedingungen wie für das erste Testszenario, jedoch wird für die Wahrscheinlichkeitsberechnung eine sequenzielle Betrachtung des Kontextinformationstyp des Ortes verwendet. Das heißt, es wird nicht nur die zuletzt eingegangene Kontextinformation des Typs „Ort“ berücksichtigt, sondern die letzten drei eingegangenen Kontextinformationen und ihre Eingangsreihenfolge werden beachtet.

Der Grund für die Auswahl dieser Testszenarien liegt darin, eine möglichst umfangreiche Bewertung der Verarbeitungskomponente und deren Logik zu ermöglichen. Durch die Unterschiede zwischen dem ersten und dem zweiten Testszenario wird der Einfluss der Wahrscheinlichkeitsberechnung auf die Erkennung einer erfolgreichen Bearbeitung erfasst. Insbesondere im Kontext von etwaigen Fehlern bei der Kontexterfassung kann dieser Einfluss eine wichtige Rolle spielen. Das dritte Testszenario zeigt hingegen den Einfluss von Kontextinformationssequenzen auf die Güte der Arbeitsschritterkennung, im Vergleich zu einer statischen Betrachtung.

Die Ergebnisse der vier Testläufe pro Testszenario werden zusammengefasst und als Auswertung in Tabelle 5.5 bis Tabelle 5.7 repräsentiert. Die Aspekte für die Auswertung sind so gewählt, dass ein Überblick über die Testläufe und deren Umfang gegeben wird. Zusätzlich werden Aspekte entsprechend der zu prüfenden Kriterien aus Tabelle 5.4 hergeleitet. Insbesondere für Kriterium K3 (Reaktion auf Kontextinformationen) werden die Aspekte „Anzahl eindeutige Arbeitsschritterkennungen“ und „Anzahl uneindeutige Arbeitsschritterkennungen“ betrachtet. Eine Bewertung der Testergebnisse findet an dieser Stelle nicht statt. Stattdessen erfolgt diese mit der Gesamtevaluation des entwickelten Systems in Abschnitt 5.6.

Tabelle 5.5: Zusammengefasste Testergebnisse für das erste Testszenario des Komponenten-tests

Testszenario 1	Testlauf 1	Testlauf 2	Testlauf 3	Testlauf 4
Anzahl der Iterationen bis Workflow-Ende	2361	2620	2076	1874
Anzahl Arbeitsschrittwechsel	91	77	125	141
Anzahl eindeutige Arbeitsschritterkennungen	349	229	454	501
Anzahl uneindeutige Erkennungen	452	279	638	650
Kausal korrekte Abfolge	Ja	Ja	Ja	Ja
Bestimmung der validen Arbeitsschritte	Ja	Ja	Ja	Ja

Tabelle 5.6: Zusammengefasste Testergebnisse für das zweite Testszenario des Komponententests

Testszenario 2	Testlauf 1	Testlauf 2	Testlauf 3	Testlauf 4
Anzahl der Iterationen bis Workflow-Ende	270	292	194	439
Anzahl Arbeitsschrittwechsel	35	25	31	35
Anzahl eindeutige Arbeitsschritterkennungen	42	53	62	52
Anzahl uneindeutige Erkennungen	79	100	54	155
Kausal korrekte Abfolge	Ja	Ja	Ja	Ja
Bestimmung der validen Arbeitsschritte	Ja	Ja	Ja	Ja

Tabelle 5.7: Zusammengefasste Testergebnisse für das dritte Testszenario des Komponententests

Testszenario 3	Testlauf 1	Testlauf 2	Testlauf 3	Testlauf 4
Anzahl der Iterationen bis Workflow-Ende	625	516	630	661
Anzahl Arbeitsschrittwechsel	65	44	62	76
Anzahl eindeutige Arbeitsschritterkennungen	219	135	224	221
Anzahl uneindeutige Erkennungen	187	70	143	150
Kausal korrekte Abfolge	Ja	Ja	Ja	Ja
Bestimmung der validen Arbeitsschritte	Ja	Ja	Ja	Ja

## 5.4 Integrationstest

Im Gegensatz zum Komponententest liegt das Ziel des Integrationstests nicht auf der Bewertung der Verarbeitungskomponente und deren prinzipieller Funktionsweise, sondern auf der Integrierbarkeit der Verarbeitungskomponente in ein bestehendes Assistenzsystem für manuelle Tätigkeiten. Entsprechend dem gewählten Anwendungsszenario erfolgt die Integration in den Handarbeitsplatz der *SmartFactory*<sup>KL</sup> den Beschreibungen der Umsetzung in Abschnitt 5.2

folgend. Um eine Bewertung der Integration ohne Einfluss von Störgrößen durchführen zu können, wird der Integrationstest in zwei Szenarien aufgeteilt. Jedes Szenario dient dazu, einen unabhängigen Teil der Integration zu überprüfen. Das erste Szenario betrifft die Integration mit dem Eingabebereich des Assistenzsystems (vgl. Erfassung in Abbildung 5.18), während das zweite Szenario die Integration mit dem Ausgabebereich (vgl. Visualisierung in Abbildung 5.19) betrachtet. Die Entkopplung der zwei Bereiche führt dazu, dass die interne Arbeits- und Funktionsweise der Verarbeitungskomponente keinen störenden Einfluss auf den Integrationstest ausübt. Die Bewertung der internen Funktionsweise erfolgt einerseits durch einen direkten Test der Funktionsweise im Komponententest (vgl. Abschnitt 5.3), andererseits durch eine indirekte Bewertung des Gesamtsystemverhaltens im Systemtest (vgl. Abschnitt 5.5). Der Aufbau und die Durchführung der Testszenarien des Integrationstests werden im Nachfolgenden beschrieben. Eine Zusammenfassung der Testergebnisse folgt am Ende dieses Kapitels in Tabelle 5.9.

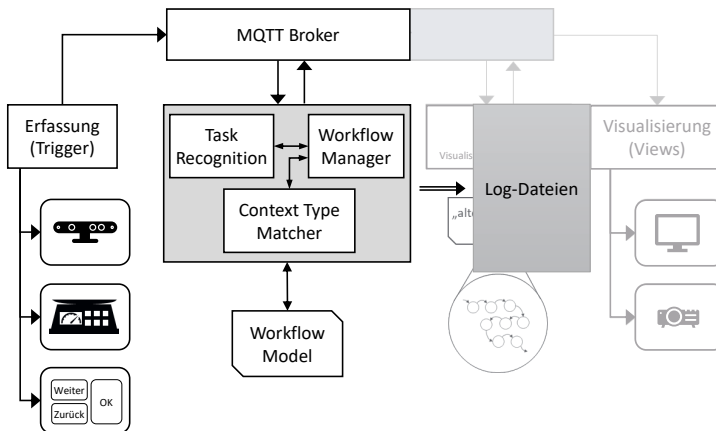


Abbildung 5.18: Systemarchitektur für das erste Testszenario des Integrationstests

### Versuchsaufbau für Testszenario 1 - Integration mit den Eingabesystemen:

Abbildung 5.18 zeigt die Systemarchitektur des Versuchsaufbaus für den Integrationstest der Verarbeitungskomponente mit den Erfassungskomponenten des *SmartFactory*<sup>KL</sup>-Assistenzsystems. Die ausgegrauten Komponenten des Ausgabebereiches und der Visualisierung sind in diesem Szenario außer Betrieb genommen. Um dennoch eine Erfassung der Reaktionen der Verarbeitungskomponente durchführen zu können, wird in Anlehnung an den Komponententest eine Log-Datei der Verarbeitungskomponente erstellt. Jede eingehende Interaktion beziehungsweise Kontextinformation führt zu einem neuen Eintrag in dieser Log-Datei. Um eine vollumfängliche Bewertung der Integration mit den Erfassungskomponenten durchzuführen, werden vier Testläufe mit jeweils zehn zufällig gewählten Interaktionen je Erfassungskomponente ausgelöst. Die ausgelösten Interaktionen werden schriftlich und in zeitlich korrekter Reihenfolge erfasst. Zusätzlich werden die über den MQTT Broker übermittelten Nachrichten gespeichert. Die



Auswertung eines Testlaufs erfolgt auf Basis der erfassten Interaktionen und der dabei generierten Log-Datei der Verarbeitungskomponente, indem diese in einer gemeinsamen Tabelle zusammengeführt werden. Eine Übersicht der möglichen Interaktionen je Erfassungskomponente gibt die Auflistung in Tabelle 5.8.

Tabelle 5.8: Übersicht der möglichen Interaktionen je Systemkomponente

	GUI	Materialbereitstellung	Tiefenbildkamera
Interaktion 1	Workflow starten	Entnahme oder Ablage von Komponenten	Ortserkennung der Hände
Interaktion 2	Workflow stoppen		Nutzung von Werkzeug
Interaktion 3	Arbeitsschritt als erledigt markieren		
Interaktion 4	Händisches Auslösen von Kontextinformationen		

**Versuchsaufbau für Testszenario 2 - Integration mit den Ausgabesystemen:**

Die Systemarchitektur für den zweiten Versuchsaufbau, als Integrationstest der Verarbeitungskomponente mit dem Ausgabebereich des Assistenzsystems, ist in Abbildung 5.19 dargestellt. Bei diesem Testszenario sind die Teilkomponenten der Verarbeitung aktiv angesteuert, sodass diese verschiedene Zustände einnimmt und entsprechend dieser Zustände Nachrichten an den Ausgabebereich übermittelt. Die Reaktionen der Visualisierungssysteme auf die Nachrichten werden durch die testende Person erfasst und schriftlich festgehalten. Die Erfassung folgt in tabellarischer Form, wobei für jede aktive Steuerung ein Eintrag zur durchgeführten Aktion, zur Reaktion der Projektion und zur monitorbasierten Visualisierung angelegt wird. Zur aktiven Steuerung der Verarbeitungskomponente werden die in Abschnitt 5.2.3 definierten MQTT-Nachrichten des MQTT-Topics „HAP/Control/#“ verwendet. Die Nachrichten werden über einen eigenständigen MQTT-Client erstellt und an den Broker übermittelt. Für das Testszenario wird ein Testlauf durchgeführt, in dem:

- 1. der Workflow des Stellantriebes geladen wird
- 2. nacheinander die Anzeige jedes legitimen Arbeitsschritts angestoßen wird
- 3. die Anzeige eines nicht legitimen Arbeitsschritts angestoßen wird
- 4. der aktuell angezeigte Arbeitsschritt als erfolgreich bearbeitet gemeldet wird
- 5. Schritt 2 bis 4 viermal wiederholt wird
- 6. Die Beendigung des Workflows angestoßen wird

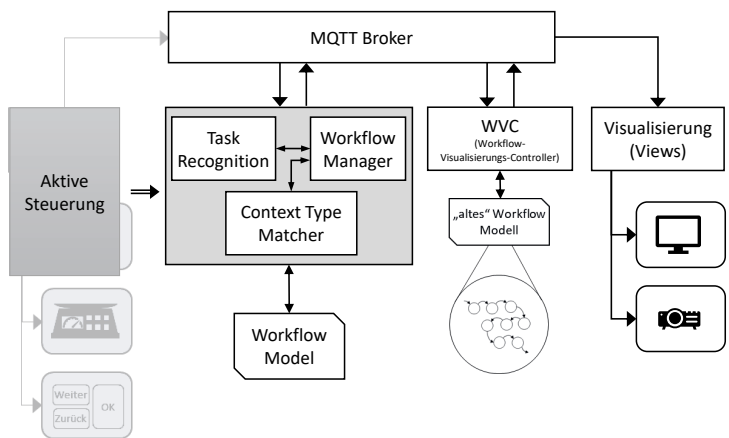


Abbildung 5.19: Systemarchitektur für das zweite Testszenario des Integrationstests

Die Auswertung der Testläufe aus den beiden Testszenarien findet durch einen Vergleich der angestoßenen Interaktion und der Reaktion aus dieser Interaktion statt. Die Ergebnisse dieses Vergleichs sind in Tabelle 5.9 aufgeführt. Für jeden Testlauf werden die Anzahl der Interaktionen, bei der eine Übereinstimmung zwischen der erwarteten /beabsichtigten Reaktion und der beobachteten Reaktion besteht, sowie die Anzahl an Interaktionen ohne Übereinstimmung aufgeführt. Eine Bewertung und Interpretation dieser Ergebnisse finden in der Gesamtevaluation in Abschnitt 5.6 statt.

Tabelle 5.9: Zusammenfassung der Ergebnisse aller Testläufe des Integrationstests

Szenario	Testlauf	Anz. der Übereinstimmungen	Anz. der Nichtübereinstimmungen
1	1	46	2
	2	44	5
	3	46	3
	4	46	2
2	1	21	0

## 5.5 Systemtest

Die Aufgabe des Systemtests ist die Verifikation und die Validierung des entwickelten kontextsensitiven, modularen Assistenzsystems (vgl. Kapitel 4). Im Gegensatz zum Komponententest und zum Integrationstest findet keine Betrachtung einzelner Teilsysteme oder der Interaktionen zwischen den Teilsystemen statt, sondern es erfolgt ein Test zur Bewertung des Gesamtsystems. Der Systemaufbau für den Testverlauf richtet sich nach den Beschreibungen und Darstellungen zu den Vorbereitungen in Abschnitt 5.1 und Abschnitt 5.2, wobei die Systemarchitektur für den Aufbau in Abbildung 5.4 dargestellt wurde. Um eine Bewertung des entwickelten Assistenzsystems aus verschiedenen Sichtweisen durchführen zu können, teilt sich der Systemtest in drei Testszenarien. Jedes Testszenario bezieht sich dabei auf eine der folgenden potenziellen Nutzergruppen:

1. Systemexperten, die das System und dessen Verhalten kennen
2. Informierte Nutzer, die eine Erklärung und Einführung in das System erhalten haben, jedoch kein tiefergehendes Detailwissen besitzen
3. Uninformierte Nutzer, die kein Wissen über das spezifische Systemverhalten besitzen

Das Vorgehen je Testszenario folgt demselben Prinzip und Aufbau. Es werden drei Testpersonen ausgewählt, die an keinem anderen Testszenario oder Testlauf teilgenommen haben. Das heißt, je Testperson wird ein Testlauf durchgeführt, wobei die Testperson die folgenden Teilaufgaben in sequenzieller Reihenfolge bearbeitet:

- Einarbeitung in den Arbeitsauftrag und die Nutzungsweise des Systems
- Starten des Workflows (der Assistenz) zur Reparatur des Stellantriebes
- Durchführung der Reparatur nach eigenen Erfahrungen und eigenem Wissen zur Reparatur von technischen Systemen sowie mithilfe der Unterstützung durch das System

Eine Abweichung von diesen Teilaufgaben besteht lediglich für die Testgruppe der informierten Nutzer, die zusätzlich zur Einführung in den Arbeitsauftrag und die Nutzungsweise des Systems auch eine Einführung in die Funktionsweise des Systems erhält. Damit eine Vergleichbarkeit aller Testläufe möglich ist, werden diese durch dieselbe Person betreut. Die Aufgaben der betreuenden Person sind die Beobachtung und Überwachung des Testlaufs, um das Nutzerverhalten der Testperson zu erfassen sowie die Einhaltung von Rahmenbedingungen sicherzustellen. Zur Dokumentation der gemachten Beobachtungen steht ein Erfassungsbogen zur Verfügung (siehe Anhang B). Die Rahmenbedingungen werden in einem Arbeitsauftrag und der Nutzungsweise des Systems angegeben. Ein weitere Aufgabe der betreuenden Person ist die händische Erzeugung von Kontextinformationen, die das System nicht selbstständig erfassen und bereitstellen kann. Im Falle des *SmartFactory*<sup>KL</sup>-Assistenzsystems sind dies die Kontextinformationen des Typs „Aktivität“ sowie die Kontextinformationen „Haube“ und „Steuerplatine“. Entsprechend Abschnitt 5.2.1 wird die Benutzungsschnittstelle der GUI verwendet, um die händische Auslö-

sung der Kontextinformationen durchzuführen. Nach Abschluss eines Testlaufs wird zusätzlich ein Fragebogen für die Testpersonen bereitgestellt, in dem Erfahrungen der Testperson und Meinungen der Testperson gesammelt werden.

Die Informationen zu den Testläufen und deren Rahmenbedingungen sowie zu den Arbeitsaufträgen stehen den Testpersonen in Form von einheitlichen Dokumenten zur Verfügung. Aufgrund der thematisch unterschiedlichen Ziele wird je ein Dokument für

- den **Arbeitsauftrag** und die Einführung in das Anwendungsszenario,
- die **Nutzungsweise** des Assistenzsystems und
- **Systembeschreibung** zur Einführung in die Funktionsweise des Assistenzsystems mit der integrierten Verarbeitungskomponente

erstellt (siehe Anhang B). Die Dokumente zum Arbeitsauftrag und der Nutzungsweise stehen dabei allen Probanden zur Verfügung, wohingegen das Dokument zur Einführung in das Assistenzsystem der Gruppe der informierten Nutzer vorbehalten ist. Die Durchführung des Systemtests und des Versuchsaufbaus sind schematisch in Abbildung 5.20 dargestellt.

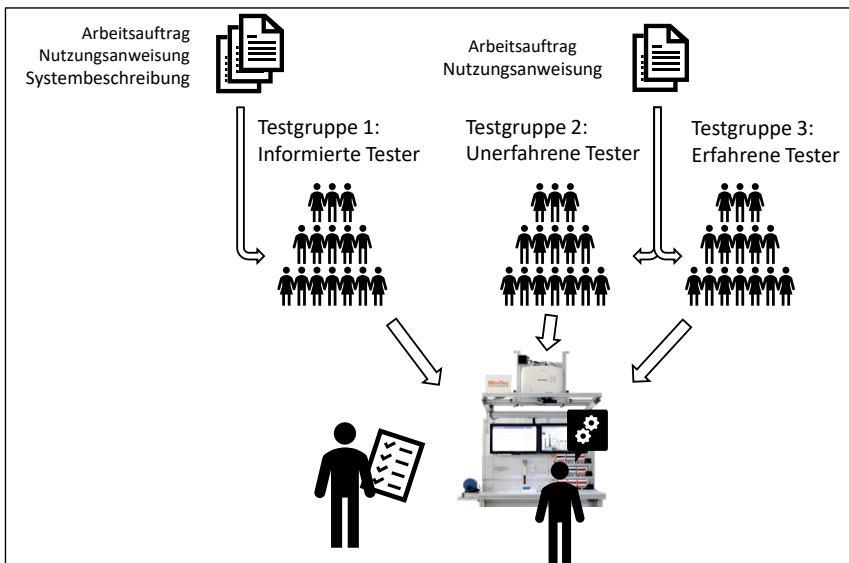


Abbildung 5.20: Schematische Darstellung des Systemtests mit drei Nutzergruppen

Eine Zusammenfassung der Ergebnisse aus den Testläufen der drei Szenarien ist am Ende des Kapitels in Tabelle 5.10 abgebildet. Testläufe des gleichen Szenarios werden dabei zusammen betrachtet, eine Aufteilung in die drei Testszenarien bleibt jedoch bestehen. Für die Zusammenfassung und Bewertung der Ergebnisse bezüglich der Erkennungsqualität wird ein Punktesystem mit einer maximalen Punktzahl von neun Punkten verwendet.

Die Punktzahl zur „Qualität der kontextsensitiven Erkennung“ wird dabei nach den folgenden Vorschriften berechnet:

- Anpassung der Punktzahl bei der Frage „Wie zeitlich passend kam die Unterstützung?“:
  - Zu früh   ⇒ -2 Punkte
  - Zu spät    ⇒ -2 Punkte
  - Passend    ⇒ +0 Punkte
- Anpassung der Punktzahl bei der Frage „Als wie gut empfinden Sie die Erkennung des bearbeiteten Arbeitsschritts?“:
  - Gut         ⇒ +3 Punkte
  - Neutral    ⇒ +2 Punkte
  - Schlecht   ⇒ +1 Punkte

Die Berechnung der Punktzahl zur „Qualität der Erkennung abgeschlossener Arbeitsschritte“ ergibt sich aus den nachstehenden Vorschriften:

- Gut         ⇒ +3 Punkte
- Neutral    ⇒ +2 Punkte
- Schlecht   ⇒ +1 Punkte

Die gesammelten und zusammengefassten Ergebnisse des Systemtests werden an dieser Stelle nicht bewertet. Stattdessen wird eine Bewertung im Zuge der Evaluation des Gesamtsystems in Abschnitt 5.6 zusammen mit den Ergebnissen des Komponententests (vgl. Abschnitt 5.3) und Integrationstests (vgl. Abschnitt 5.4) durchgeführt.

Tabelle 5.10: Zusammenfassung der Ergebnisse des Systemtests

	Informierte Nutzer	Unerfahrene Nutzer	Systemexperten
<b>Erfolgreiche Durchführungen des Arbeitsauftrages</b>	3/3	3/3	3/3
<b>Anzahl kontextsensitiver Erkennungen</b>	9	19	12
<b>Anzahl aktiver Nutzungen der Kontextsensitivität</b>	7	14	11
<b>Qualität der kontextsensitiven Erkennung</b>	8/9	5/9	6/9
<b>Qualität der Erkennung abgeschlossener Arbeitsschritte</b>	8/9	8/9	8/9
<b>Wünsche nach eigenständigem Abschluss der Arbeitsschritte</b>	0/3	0/3	1/3

## 5.6 Bewertung und Evaluation des Gesamtsystems

Nachdem in Abschnitt 5.2 die modellierte Verarbeitungskomponente umgesetzt und in das Assistenzsystem der *SmartFactory*<sup>KL</sup> integriert wurde, ist das entwickelte System getestet worden. Dabei wurde das System in Bezug auf die Arbeitsweise der Verarbeitungskomponente (vgl. Abschnitt 5.3), die Integrierbarkeit in bestehende Systeme (vgl. Abschnitt 5.4) sowie die Funktionsweise des gesamten kontextsensitiven, modularen Assistenzsystems (vgl. Abschnitt 5.5) untersucht. Aufbauend auf den dabei gesammelten Testergebnissen wird das System in diesem Kapitel bewertet und evaluiert. Hierzu werden im ersten Schritt die Ergebnisse des Komponententests, des Integrationstests sowie des Systemtests vorgestellt und erläutert. Nachdem eine Bewertung für alle Tests durchgeführt wurde, erfolgt eine Prüfung der Zielerreichung des entwickelten kontextsensitiven, modularen Assistenzsystems auf Basis der globalen Anforderungen (vgl. Abschnitt 3.3).

### Bewertung des Komponententests

Das Ziel des Komponententest ist es, die Funktionalität der Verarbeitungskomponente mit Bezug zu den funktionalen Anforderungen zu überprüfen. Hierzu zählen vor allem die Anforderungen hinsichtlich einer kontextsensitiven Erkennung eines Arbeitsschritts (vgl. **GA5**, **GA9**, **LA14**, **LA15**, **LA20**), die Bereitstellung einer Meldung zum erkannten Arbeitsschritt (vgl. **LA3**, **LA9**) sowie die Wahrung der kausalen Bearbeitungsreihenfolgen (vgl. **GA10**, **LA12**, **LA13**, **LA16**, **LA19**). Die Bewertung dieser Funktionsweise erfolgt auf Basis der Beschreibungen des Komponententests und mithilfe von drei Testszenarien, die anhand von sieben Kriterien ausgewertet werden. Die Unterteilung in drei Testszenarien dient dazu, um nicht nur die Einhaltung der Anforderungen, sondern auch die Qualität der entwickelten Logik zu prüfen. Durch Veränderungen am Workflow-Modell und an der Verarbeitungslogik zwischen den drei Testszenarien erfolgt zusätzlich eine Überprüfung der Anpassbarkeit und Modifizierbarkeit des Systems (vgl. **GA2**, **GA3**, **GA4**, **LA4**, **LA5**).

Eine Auswertung der Testergebnisse zeigt, dass die sieben Kriterien erfüllt werden und somit alle Anforderungen eingehalten werden, die an die Verarbeitungskomponente gestellt werden. Jedoch offenbaren sich deutliche Unterschiede in der Qualität der kontextsensitiven Erkennung von Arbeitsschritten zwischen den drei gewählten Testszenarien. Insbesondere in Testszenario 1 (57 % aller Erkennungsversuche) und in Testszenario 2 (65 % aller Erkennungsversuche) konnte die Verarbeitungskomponente häufig keine eindeutige Bestimmung des sich in Bearbeitung befindlichen Arbeitsschritts durchführen. Da das System bei einer nicht eindeutigen Erkennung keine Veränderung des erkannten Arbeitsschritts durchführt, besteht die Möglichkeit, dass der nutzenden Person keine geeignete Unterstützung für den durchgeführten Arbeitsschritt zur Verfügung gestellt wird. Da die Testreihen mit einem zufallsbasierten Testprogramm durchgeführt wurden, lässt sich darüber jedoch keine sichere Aussage machen, da nicht bekannt ist in wie vielen der uneindeutigen Situationen tatsächlich ein Wechsel des erkannten Arbeits-

schritts notwendig gewesen wäre. Aufgrund der hohen Anzahl an Situationen besteht aber ein entsprechendes Risiko, dass es zu häufigen Fehlerkennungen kommt. Eine Verringerung des Risikos durch eine bessere Erkennung der Schritte scheint jedoch durch eine erhöhte Anzahl an betrachteten Kontextinformationen beziehungsweise durch die Betrachtung von Aktivitätssequenzen möglich zu sein. Bei einer entsprechenden Anpassung der Betrachtung in Testszenario 3 konnte die Häufigkeit einer uneindeutigen Erkennung bereits auf 41 % der Fälle reduziert werden. Welche Erkennungsrate durch das System erreicht werden muss, um ein sicheres Wechseln zwischen den erkannten Arbeitsschritten zu ermöglichen, lässt sich nach Ansicht des Autors nur durch eine Evaluation des Systems mit paraktischen Anwendern aus dem Tätigkeitsbereich der Reparatur und Nacharbeit herausfinden. Bei der Durchführung von Anwendertests kann zusätzlich erfasst werden, wann ein Wechsel des erkannten Arbeitsschritts notwendig ist.

Da aus den genannten Gründen keine sichere Aussage zur Qualität der Arbeitsschritterkennung getroffen werden kann, ist ferner keine aussagekräftige Bewertung des besten Verfahrens zur automatisierten Erkennung eines erfolgreich bearbeiteten Arbeitsschritts möglich. Zwar zeigen sich deutliche Unterschiede zwischen Testszenario 1 (durchschnittlich 2233 Iterationen) und Testszenario 2 (durchschnittlich 299 Iterationen), diese deuten jedoch lediglich darauf hin, dass die Arbeitsschritte in Szenario 2 nach weniger Iterationen als erledigt erkannt wurden. Eine frühere Erkennung kann aber bedeuten, dass der Arbeitsschritt als erledigt markiert wird, obwohl dieser noch nicht durch die bearbeitende Person abgeschlossen wurde. In diesem Fall würde das System jedoch keine Hilfestellung mehr für diesen Arbeitsschritt bereitstellen können. Bei einer späteren Erkennung mit 100 % Übereinstimmung zwischen dem erfassten Kontext und der optimalen Beobachtung besteht jedoch die Gefahr, dass der Arbeitsschritt aufgrund einer fehlerhaften Kontexterfassung gar nicht als erledigt markiert wird. Zur Bewertung der automatisierten Erkennung eines bearbeiteten Arbeitsschritts ist folglich eine Untersuchung mit Anwendern des Systems notwendig.

Da die an das System gestellten Anforderungen erfüllt werden und eine Optimierung der Erkennungsqualität nicht zur Zielsetzung dieser Arbeit gehört, werden im weiteren Verlauf keine weitere Untersuchungen oder Evaluationen der Erkennungsqualität durchgeführt.

### **Bewertung des Integrationstests**

Mithilfe des Integrationstests wird überprüft, ob das entwickelte System in ein bestehendes Assistenzsystem integriert werden kann (vgl. **GA1**) und ob die Ein- und Ausgangssignale korrekt interpretiert werden (vgl. **LA2, LA3, LA8**). Hierzu wird durch zwei Testszenarien eine partielle Integration in das System durchgeführt, wodurch einerseits eine Überprüfung der reinen Integrierbarkeit ohne Einflussnahme der Verarbeitungslogik und andererseits eine plattformübergreifende und systemunabhängige Integration (vgl. **GA8, LA6, LA7, LA8**) erreicht wird. Die Integration der Systemkomponente wird durch das eventbasierte Kommunikationsprotokoll MQTT umgesetzt (vgl. **LA1**).

In Testszenario 1 fand eine Integration des Systems mit Systemkomponenten zur Erfassung von Kontextinformationen und Steuersignalen statt. Die Auswertung der dabei gesammelten Testergebnisse verdeutlicht, dass die Integration des Systems problemlos möglich ist. Kontextinformationen von allen Erfassungskomponenten wurden zum Zeitpunkt der Erfassung an die Verarbeitungskomponente übermittelt und durch diese empfangen. Die Verwendung von unterschiedlichen Erfassungskomponenten (GUI, Intelligente Materialbereitstellung, Tiefenbildkamera), die jeweils auf einem eigenen System implementiert sind, zeigt dabei, dass die Integration unabhängig von der spezifischen Komponente sowie deren Plattform durchführbar ist. Werden zeitgleiche mehreren Kontextinformationen erfasst, zum Beispiel bei der Ablage von Komponenten, so können diese ebenso durch die Verarbeitungskomponente empfangen und verarbeitet werden. Hinsichtlich einer Bewertung der erfassten und übermittelten Kontextinformationen zeigen die Testergebnisse, dass in etwa 6 % der Fälle keine Erfassung der ausgelösten Kontextinformation stattgefunden hat. Dies lässt sich jedoch auf die Qualität der Erfassungskomponenten zurückführen und ist unabhängig von der Integration oder der Verarbeitungskomponente. Nichtsdestotrotz wurde dieser Versuch im Zuge des Integrationstests durchgeführt, um die Nutzbarkeit des Gesamtsystems für die Durchführung von Anwendertests zu überprüfen (vgl. Systemtest in Kapitel 5.5). Da die Fehlerkennung deutlich unter 10 % liegt und die Anwendertests zusätzlich durch eine betreuende Person überwacht werden, die eine händische Erkennung von Kontextinformationen durchführen kann, wird das System als geeignet angesehen. Für einen tatsächlichen Betrieb des Gesamtsystems sollte jedoch eine Verbesserung der Erfassungsgenauigkeit realisiert werden.

In Testszenario 2 fand eine Integration mit den Systemkomponenten der Ausgabe und Visualisierung des bestehenden Systems statt. Hierzu wurde die Verarbeitungskomponente über Steuersignale in spezifische Situationen versetzt. Anhand der Ausgabekomponenten wurde gleichzeitig überprüft, ob die Verarbeitungskomponente eine korrekte Reaktion und Ansteuerung der Visualisierung auslösen konnte. Die Auswertung der Testergebnisse dieses Szenarios zeigt, dass auch hierbei eine problemlose Integration des entwickelten Systems möglich ist und dass in 100 % der Fälle eine Übereinstimmung zwischen dem provozierten Verhalten und dem gezeigten Verhalten stattgefunden hat.

Innerhalb der Rahmenbedingungen dieser Arbeit und des definierten Anwendungsfalles konnte mithilfe des Integrationstests gezeigt werden, dass alle Anforderungen hinsichtlich der Integrierbarkeit in ein plattform- und systemunabhängiges, modulares System erfüllt werden. Um eine allgemeingültige Aussage zur Integrierbarkeit des Systems machen zu können, sind jedoch weiterführende Untersuchungen mit anderen bestehenden Assistenzsystemen für die Unterstützung manueller Tätigkeiten notwendig.

### **Bewertung des Systemtests**

Der Systemtest verfolgt das Ziel eines abschließenden Gesamttests zur Überprüfung des geplanten Systemverhaltens und der Nutzbarkeit des Systems. Der Systemtest wird daher mit Anwen-



dern durchgeführt, um dessen Fähigkeit zur Unterstützung einer selbstbestimmten Arbeitsweise der nutzenden Person (vgl. **GA9**) zu bewerten. Ferner wird durch die Anwendertests beurteilt, ob die Anforderungen zur Umsetzung einer intuitiven und gebrauchstauglichen Benutzerschnittstelle (vgl. **GA5**, **GA6**) erfüllt werden. Hierzu werden die Anwender in drei Testgruppen aufgeteilt, je nach Wissensstand bezüglich der Funktionsweise des entwickelten Systems.

Die Auswertung der durchgeführten Anwendertests zeigt, dass die Verhaltensweise des Systems den geforderten Anforderungen entspricht und eine selbstbestimmte Bearbeitung der Arbeitsschritte bei gleichzeitiger Wahrung kausaler Abhängigkeiten zwischen Arbeitsschritten möglich ist (vgl. **GA9**, **GA10**). Alle durchgeführten Testläufe konnten von Start des Prozesses bis Abschluss der Reparatur bearbeitet werden. Der Grad an Selbstbestimmtheit während der Durchführung unterscheidet sich dabei deutlich zwischen den einzelnen Anwendern, wobei sich Tendenzen zwischen den jeweiligen Testgruppen abzeichnen. So hat die Testgruppe der Uninformierten am häufigsten eine eigene Entscheidung hinsichtlich des Bearbeitungsweges getroffen, statt den Anzeigen des Systems zu folgen. Die teilnehmenden Personen dieser Gruppe waren primär auf das Produkt und den Prozess konzentriert und haben das Assistenzsystem lediglich für eine geringe Anzahl an Hilfestellungen beachtet. Die Gruppe der Informierten hingegen ist sehr häufig dem vorgeschlagenen Bearbeitungsweg gefolgt und war eher auf das Assistenzsystem als auf das eigentliche Produkt fokussiert. Dies wäre für ein finales System in der Hinsicht schlecht, da weder ein optimaler Bearbeitungsweg ermittelt werden kann, noch das finale Bearbeitungsziel für alle Situationen bekannt ist. Eine Vermutung, weshalb es zum beobachteten Verhalten der Testgruppe der Informierten kam, liegt darin, dass die Personen mit zu vielen Informationen hinsichtlich Arbeitsauftrag, Systemverhalten und Nutzungsanweisung für das System überreizt wurden. Da der durchgeführte Systemtest unter Laborbedingungen und mit einer kleinen Nutzergruppe durchgeführt wurde, kann dieser jedoch nicht als repräsentative Nutzerstudie betrachtet werden.

Die fragebogenbasierte Beurteilung durch die Testpersonen offenbart, dass die Qualität der automatisierten und kontextsensitiven Erkennung des zu bearbeitenden Arbeitsschrittes sowie der erfolgreichen Bearbeitung eines Arbeitsschrittes vornehmlich als gut angesehen wird. Hierzu muss jedoch gesagt werden, dass durch die Verwendung der Benutzerschnittstelle der Systemtest nach den Prinzipien eines Wizard-of-Oz-Experimentes durchgeführt wurde. Das heißt, ein Teil der Reaktionen des Systems wurde durch die betreuende Person hervorgerufen. Entsprechend den Beschreibungen in Kapitel 6.2.1 betrifft dies die Bereitstellung der Kontextinformationen, die das System nicht selbstständig generieren kann. Ferner wurde auch die automatisierte Erkennung einer erfolgreichen Bearbeitung eines Arbeitsschrittes durch die betreuende Person durchgeführt. In vorgelagerten Probetestläufen hat sich gezeigt, dass das entwickelte System diese Erkennung aufgrund nicht einwandfrei funktionierender Erfassungssysteme nicht in geeigneter Weise durchführen konnten. Diese System liegen jedoch außerhalb des betrachteten Systemraums, sodass diese Beobachtung nicht in die Bewertung des entwickelten Systems

einfließt. Dasselbe gilt für Anmerkungen der Testpersonen bezüglich der visuellen Darstellung von Arbeitsanweisungen, die ebenfalls außerhalb des Betrachtungsraumes dieser Arbeit liegt. Zwar wurde von allen Testpersonen angegeben, dass die Qualität der kontextsensitiven Erkennung vornehmlich gut war, jedoch wird diese positive Beurteilung aufgrund einer suboptimalen Visualisierung nicht direkt am Gesamtsystem sichtbar und konnte nur im Gespräch ermittelt werden. Dies liegt daran, dass die Visualisierung das primäre Objekt zur Beurteilung der Erkennungsqualität ist und es somit für die Testpersonen schwierig zu unterscheiden ist, welche Auswirkungen durch das entwickelte System und welche Auswirkungen durch die Visualisierung hervorgerufen werden. Die Erkenntnisse aus dem Systemtest verdeutlichen, dass im Rahmen dieser Arbeit die Anforderungen bezüglich einer intuitiven Benutzerschnittstelle (vgl. **GA5**, **GA6**) als erfüllt angesehen werden können, wobei dieses Ergebnis nicht als repräsentativ für den Realbetrieb in der Produktionsumgebung gewertet werden kann.

**Bewertung des entwickelten Systems**

Die Bewertung des entwickelten kontextsensitiven, modularen Assistenzsystems erfolgt anhand der dargestellten Auswertung der durchgeführten Testreihen in diesem Kapitel. Hierbei zeigt sich, dass alle an das System gestellten globalen Anforderungen erfüllt wurden (vgl. Abbildung 5.21). Dies trifft insbesondere auf die Anforderungen **GA3**, **GA4**, **GA7**, **GA8**, **GA9** und **GA10** zu, die im Rahmen dieser Arbeit vollständig überprüft wurden. Da im Rahmen dieser Arbeit jedoch keine Untersuchung des Systems in Bezug auf die Integrierbarkeit in eine Vielzahl unterschiedlicher bestehender Systeme oder eine Übertragung in andere Anwendungsbereiche durchgeführt wurde, werden die Anforderungen **GA1** und **GA2** lediglich als teilweise erfüllt gewertet. Aufgrund der Integration mit unterschiedlichen und plattformunabhängigen Systemkomponenten wurde jedoch gezeigt, dass eine Integration in verschiedene Systeme möglich ist. Weiterhin sorgt der modulare Aufbau nach den SoC-Prinzipien dafür, dass Teilbereiche des Systems austauschbar und individuell nutzbar sind, ohne andere Teile des Systems verändern zu müssen. Insbesondere in Testszenario 2 des Integrationstests wurde durch die direkte Ansteuerung des *Workflow Managers* gezeigt, dass dieser losgelöst von den

ID	Anforderung	Erfüllung
GA1	Integrierbarkeit in bestehende Systeme	○
GA2	Übertragbarkeit und Wiederverwendbarkeit	○
GA3	Modifizierbarer Systemaufbau	●
GA4	Anpassbares Systemverhalten	●
GA5	Intuitive Bedienbarkeit	○
GA6	Gebrauchstauglichkeit der Bedienoptionen	○
GA7	Plattformunabhängige Systemgestaltung	●
GA8	Plattformunabhängige Integration	●
GA9	Selbstbestimmung der Nutzer	●
GA10	Wahrung kausaler Abhängigkeit	●

● erfüllt                      ○ teilweise erfüllt

Abbildung 5.21: Erfüllbarkeit der globalen Anforderungen durch das entwickelte System

anderen Komponenten verwendet werden kann. Zusätzlich zu den Anforderungen **GA1** und **GA2** werden die Anforderungen **GA5** und **GA6** lediglich als teilweise erfüllt bewertet. Dies liegt vor allem daran, dass die Bewertung einer intuitiven und gebrauchstauglichen Bedienbarkeit eines Systems von subjektiver Natur ist. Eine adäquate Bewertung ist daher nur durch eine Nutzerstudie mit einer Vielzahl von Anwendern möglich. Da eine solche Studie den Zugang zu einer Vielzahl an praktischen Nutzern sowie entsprechenden Unternehmen benötigt um aussagekräftige Erkenntnisse erzielen zu können, konnte dies im Rahmen der Möglichkeiten dieser Arbeit nicht umgesetzt werden (vgl. Implikationen für Wissenschaft und Praxis in Abschnitt 6.1). Dennoch konnte mithilfe der durchgeführten Anwendertests gezeigt werden, dass eine intuitive und gebrauchstaugliche Bedienung in ihren Grundzügen erreicht wurde. Für eine Optimierung ist daher keine komplette Neuentwicklung des Systems notwendig. Stattdessen kann eine Adaption durch den modularen und anpassbaren Systemaufbau erfolgen.

## 6 Zusammenfassung und Ausblick

Die immer stärkere Nachfrage nach kundenindividuellen Produkten sorgt bei produzierenden Unternehmen für eine größere Bandbreite an Produkttypen und -variationen bis hin zu völlig individuellen Produkten, die als Sonderlösungen entwickelt und gebaut werden. Hierfür werden flexiblere Produktionssysteme aber auch Unterstützungssysteme notwendig sein, um Fachkräften alle Informationen zu einem Produkt oder Produktionsauftrag zur Verfügung zu stellen. Insbesondere für den Bereich der Reparatur von technischen Systemen sowie zur Ermöglichung einer selbstbestimmten Arbeitsweise der Fachkräfte sind geeignete Unterstützungssysteme notwendig, die sich flexibel an die nutzende Person und ihre Arbeitswege anpassen. Durch solche Systeme können die nutzenden Personen die Bearbeitung eines Arbeitsauftrages nach eigene Entscheidungen und Vorlieben durchführen.

Möglich wird diese Anpassungsfähigkeit durch eine kontextsensitive Erfassung der Arbeitssituation in Verbindung mit Informationen zum durchzuführenden Arbeitsauftrag. Da viele Produkte durch ihren konstruktionsbedingten Aufbau nicht in jeder beliebigen Art und Weise montiert oder repariert werden können, muss die flexible Anpassungsfähigkeit des Systems auf tatsächlich durchführbare Arbeitsschritte beschränkt werden. Aus diesem Grund müssen mögliche Bearbeitungswege für jedes Produkt beziehungsweise Arbeitsauftrag entsprechend modelliert und dem Assistenzsystem zur Verfügung gestellt werden. Soll die Unterstützung gar für stark veränderliche Arbeitsplätze, wie sie bei Reparaturaufgaben häufig anzutreffen sind, geeignet sein, muss dies bei der Modellierung des Arbeitsprozesses zusätzlich beachtet werden. Ferner muss entweder das verwendete Assistenzsystem an die veränderte Situation angepasst werden oder die modellierten Arbeitsaufträge müssen einfach auf ein anderes System übertragen werden können.

Diese Arbeit beschäftigt sich daher mit der Entwicklung eines modularen, kontextsensitiven Assistenzsystems, das eine Anpassung an veränderliche Arbeitssituationen ermöglicht sowie mit der Modellierung von flexibel bearbeitbaren Arbeitsplänen zur Nutzung in einem solchen Assistenzsystem. Die Grundlage hierzu lieferte eine umfassende Recherche und Analyse des Stands der Wissenschaft und Technik in Kapitel 2. Fokus der Recherche lag auf den zur Zeit dieser Arbeit existierenden Assistenzsystemen für manuelle Tätigkeiten (Abschnitt 2.1), den Grundlagen zur semantischen Modellierung von Arbeitssituationen (Abschnitt 2.2) sowie den Vorgehensweisen zur Modellierung von Arbeitsprozessen (Abschnitt 2.3).

Damit eine sukzessive und nachvollziehbare Entwicklung des Assistenzsystems auf dieser Basis möglich ist, erfolgte im ersten Schritt die Beschreibung der Vorgehensweise (Kapitel 3). Die Kernaspekte davon sind die Festlegung eines geeigneten methodischen Vorgehens (Abschnitt 3.1), die Abgrenzung des betrachteten Systemraums (Abschnitt 3.2) sowie die Definition von globalen Anforderungen an die Entwicklung (Abschnitt 3.3).

Der methodischen Vorgehensweise folgend wurde im zweiten Schritt eine umfangreiche Anforderungsanalyse anhand bestehender Assistenzsysteme durchgeführt (Abschnitt 3.4). Ziel dieser Analyse ist es, sicherzustellen, dass das entwickelte System in bestehende Systeme integriert werden kann, um dort als dessen Kernkomponente ein modulares, kooperierendes und kontextsensitives Gesamtsystem zu realisieren. Die Durchführung orientierte sich an den Prinzipien des Software Reverse Engineerings (SRE), indem die bestehenden Systeme als zusammengefasstes, abstrahiertes Modell repräsentiert werden und anschließend eine Herleitung von Anforderungen auf Basis dieses Modelles durchgeführt wurde. Dem SRE entsprechend unterteilt sich dieses Vorgehen in eine detaillierte Betrachtung der Verarbeitungskomponente (Abschnitt 3.4.1), der Systemarchitektur (Abschnitt 3.4.2) sowie des Gesamtsystemverhaltens (Abschnitt 3.4.3). Den Abschluss der Analysephase bildet die Zusammenfassung aller hergeleiteten funktionalen und lokalen Anforderungen (Abschnitt 3.4.4).

Aufbauend auf den lokalen und den globalen Anforderungen wurde im dritten Schritt die Modellbildung des kontextsensitiven, modularen Assistenzsystems durchgeführt (Kapitel 4). Hierzu wurde eine Beschreibung des geplanten Zielsystemverhaltens erstellt (Abschnitt 4.1) und die relevanten Kernkomponenten zur Umsetzung dieses Verhaltens abgeleitet. Die Komposition dieser Komponenten zu einem Gesamtsystem wurde zusammen mit der Modellierung der Interaktionen zwischen diesen Komponenten in der Systemarchitektur durchgeführt (Abschnitt 4.2). Die Hauptaufgabe der Modellbildung ist die Entwicklung des Arbeitsplanmodells (Abschnitt 4.3.4) sowie die darauf aufbauende kontextsensitive und flexible Erkennung der Arbeitssituation (Abschnitt 4.3.3) bei gleichzeitiger Beachtung der konstruktionsbedingten Bearbeitungswege (Abschnitt 4.3.2). Ferner wurden die Teilsystemkomponenten konzipiert, welche die Einzelaufgabe der kontextsensitiven Erkennung und der Verwaltung der Bearbeitungswege erfüllen (Abschnitt 4.3.5).

Die praktischen Untersuchung und Evaluation der Entwicklung (Kapitel 5) erfolgte zusammen mit der Umsetzung und Implementierung des modellierten Systems (Abschnitt 5.2). Die Grundlage hierfür bietet das Anwendungsszenario einer Reparatur und das bestehende Assistenzsystem der *SmartFactory*<sup>KL</sup>, in welches das entwickelte System integriert wurde (Abschnitt 5.1). Damit eine umfangreiche Untersuchung verschiedener Teilaspekte des Systems ermöglicht wurde, untergliedert sich die Untersuchung in drei individuelle Testszenarien. Im Komponententest wurde die Funktionsweise der entwickelten Kernkomponenten hinsichtlich der kontextsensitiven Erkennung der Arbeitssituation und der Verwaltung des Bearbeitungsprozesses analysiert (Abschnitt 5.3). Der Integrationstest diente der Überprüfung, ob das entwickelte System ein-

fach und vollständig in bestehende Systeme integriert werden kann (Abschnitt 5.4). Im finalen Systemtest wurde das Gesamtsystemverhalten mithilfe von praktischen Anwendertests (Abschnitt 5.5) untersucht. Aufbauend auf den gesammelten Anforderungen (Abschnitt 5.6) wurden anschließend die drei Testszenarien ausgewertet und das System evaluiert.

## 6.1 Implikationen für Wissenschaft und Praxis

Durch die Eingrenzung des betrachteten Systemraums und das damit einhergehende Nichtbeachten von Frage- und Problemstellungen, aber auch durch gesammelte Erfahrungen während dieser Arbeit, ergeben sich weiterführende Anknüpfungspunkte an diese Arbeit. Durch die wissenschaftliche und praktische Bearbeitung der damit verknüpften Problemstellungen sind Verbesserungen des entwickelten Systems möglich, die hin zu einem vollständig praxistauglichen System führen. Im Folgenden werden die identifizierten Anknüpfungspunkte näher erläutert:

**Optimierung der kontextsensitiven Arbeitsschritterkennung:** Um eine stabile Funktionsweise des Assistenzsystems zu erreichen, die den nutzenden Personen in jeglichen Situationen unterstützende Informationen und Arbeitsanweisungen bereitstellt, muss eine Untersuchung der optimalen Arbeitsschritterkennung durchgeführt werden. Im Rahmen dieser Arbeit konnte gezeigt werden, dass die Prinzipien der kontextsensitiven Erkennung funktionsfähig sind, die Qualität der Erkennung jedoch maßgeblich durch das semantische Kontextinformationsmodell sowie die darauf aufbauende Bewertungslogik beeinflusst wird. Ohne weiterführende Studien und Versuche, unter anderem in realen Produktionsumgebungen, ist eine adäquate Erkennung der Arbeitsschritte und somit eine Unterstützung für die nutzenden Personen nicht gewährleistet. Dies kann zu Misstrauen und Nichtbenutzen des Assistenzsystems führen, wodurch die positiven Effekte eines solchen Systems ungeutzt bleiben.

**Optimierung des Arbeitsprozessmodells und Verkettung von Arbeitsschritten:** Ein Ziel dieser Arbeit war die Kombination deterministischer und probabilistischer Prozesse in einem gemeinsamen Arbeitsprozessmodell. Für nutzende Personen mit einem hohen Unterstützungsbedarf kann die starke Trennung dieser beiden Bereiche jedoch zu einer ineffizienten Bearbeitung des Arbeitsprozesses führen, wenn die Person alle Arbeitsanweisungen in der gegebenen Reihenfolge bearbeitet. Durch die starke Trennung erfolgt eine nahezu zufällige Anzeige des nächsten Arbeitsschritts sobald ein Arbeitsschritt beendet wurde. Um eine bessere Unterstützung für dieses Szenario zu erreichen, ist die Gewichtung oder Priorisierung von Teilbearbeitungswegen denkbar, sodass nach einer erfolgreichen Bearbeitung der Schritt ausgewählt wird, der am wahrscheinlichsten folgt. Ein denkbarer Ansatz hierfür ist die Erweiterung der HMM-basierten Ansätze, zum Beispiel durch die vollständige Integration von HMM, die für den jeweils aktuellen Arbeitsprozesszustand automatisiert generiert werden.

**Integration mit verschiedenen Konstellationen an Erfassungssystemen:** Durch den modularen Aufbau des entwickelten Systems haben die Qualität und die Kombination der verwendeten Erfassungssysteme einen maßgeblichen Einfluss auf die korrekte Funktionsweise des Systemverhaltens. Fehlerkennungen spiegeln sich direkt in der kontextsensitiven Arbeitsschritterkennung und der Erkennung von erfolgreich bearbeiteten Schritten wider. Ferner besteht die Möglichkeit, dass mehrere Erfassungsgeräte die gleichen Kontextinformationstypen bereitstellen und sich mit erfassten Kontextinformationen widersprechen oder bestätigen. Durch die Ausgrenzung des Bereiches der Erfassungssysteme fehlen bisher konkrete Ansätze, die mit Doppelinformationen oder Fehlerkennungen adäquat umgehen. Denkbar sind die Einführung von Erkennungsqualitätsstufen und die Einbeziehung dieser Qualitätsstufen in die Verarbeitungslogik des Systems, um eine Bewertung des Systemverhaltens für verschiedene Erfassungssystemkonstellationen ermöglichen zu können.

**Untersuchung bekannter Visualisierungsmethoden und -systeme hinsichtlich des Systemverhaltens:** Bestehende Assistenzsysteme für manuelle Tätigkeiten konzentrieren sich bisher vornehmlich auf rein deterministische oder rein probabilistische Prozessmodelle zur Unterstützung der nutzenden Person. Insbesondere in Hinblick auf das veränderte und kombinierte Prozessverhalten, das deterministische und probabilistische Aspekte beinhaltet, und der Ausgrenzung des Bereiches der Visualisierung, steht eine Untersuchung bestehender Visualisierungsansätze hinsichtlich der Nutzbarkeit für das entwickelte System aus. So kann es durch die automatisierte Erkennung und Umschaltung zwischen Arbeitsschritten zu Sinnesüberreizungen kommen. Gleichzeitig besteht die Möglichkeit, dass durchführbare Arbeitsschritte nicht klar ersichtlich sind und somit keine Unterstützungsleistung geboten wird.

**Integration in bestehende Systeme des Produktionsumfelds und Evaluation der Praxis-tauglichkeit:** Die Ergebnisse im Rahmen dieser Arbeit ergeben sich aus einer prototypischen Umsetzung und Integration eines kontextsensitiven, modularen Assistenzsystems. Zwar ist ein Machbarkeitsnachweis dadurch erbracht, die vollumfängliche Integration in Unternehmen des Produktionsumfelds steht jedoch weiterhin aus. So ist beispielsweise keine vollständige Betrachtung aller zur Integration relevanter Systeme erfolgt. Ein Beispiel hierfür ist die automatisierte Generierung und Bereitstellung von Workflows zum Beispiel durch CAD-basierte Methoden, die relevante Arbeitsschritte und Bearbeitungswege aus dem Konstruktionsmodell ableiten und in ein Workflow-Modell überführen. Ferner erfordert die Umsetzung eines marktreifen Produktes die Berücksichtigung von umfänglichen Test- und Integrationsphasen, um Probleme, instabile Funktionen und Teilkomponenten zu identifizieren.

**Überführung in andere Anwendungsdomänen und Arbeitsbereiche:** Der fokussierte Anwendungsbereich dieser Arbeit ist die Reparatur von technischen Geräten und die besonderen Ansprüche, die sich daraus an Assistenzsysteme für manuelle Tätigkeiten ergeben. Den-

noch existieren einige Anknüpfungspunkte, beispielsweise zur manuellen Montage oder dem Recycling von technischen Geräten. Eine Überführung des entwickelten Konzeptes oder von Teilaspekten in diese Anwendungsbereiche könnte daher zu Optimierungen bestehender Assistenzsysteme oder Entwicklung neuer Ansätze beitragen. Zusätzlich können die veränderten Ansprüche aus diesen Bereichen zu Verbesserungen an dem entwickelten kontextsensitiven, modularen System führen, die wiederum in den Bereich der Reparatur technischer Systeme zurückgespiegelt werden können.

Der Aufbau auf Basis unabhängiger Teilkomponenten ermöglicht eine weiterführende Betrachtung und Anpassung des Systems. In eigenständigen Forschungsvorhaben können somit die identifizierten Aspekte, aber auch davon unabhängige Ansätze, mithilfe des Assistenzsystems bearbeitet und evaluiert werden.

## 6.2 Abschließende Bewertung

Die abschließende Bewertung dieser Arbeit erfolgt anhand der identifizierten wissenschaftlichen Forschungsfragen der festgelegten Zielsetzung. Zur Beantwortung dieser Fragen wurde als übergeordnete Entwicklung die Architektur und der Aufbau eines kontextsensitiven, modularen Assistenzsystems erarbeitet (vgl. Kapitel 4). Diese Architektur erlaubt die Bearbeitung von deterministischen Arbeitsprozessen in Kombination mit einer kontextsensitiven Erkennung von Arbeitsschritten (vgl. Forschungsfrage 3).

### Forschungsfrage 3:

Wie muss die Architektur des Systems aufgebaut sein, um die Bearbeitung des Arbeitsprozesses (1) und die Erkennung des aktuellen Arbeitsschrittes (2) zu ermöglichen?

Das Kernelement dieser Entwicklung bildet das Workflow-Modell, indem ein Arbeitsprozess mit all seinen Arbeitsschritten sowie den kausalen, konstruktionsbedingten Verknüpfungen zwischen diesen Arbeitsschritten abgebildet wird (vgl. Abschnitt 4.3.4). Durch dieses Modell kann das System zu jedem Zeitpunkt des Bearbeitungsprozesses die durchführbaren Bearbeitungswege und Arbeitsschritte identifizieren, wodurch flexibel zwischen Arbeitsschritten hin und her gewechselt werden kann. Für die Verarbeitung des Workflow-Modells wurden unabhängige Teilkomponenten entwickelt (vgl. Abschnitt 4.3.5), die die Bereiche der Arbeitsprozessverwaltung und der Erkennung des in Bearbeitung befindlichen Arbeitsschrittes umsetzen. (vgl. Forschungsfrage 1).

### Forschungsfrage 1:

Wie kann der Arbeitsprozess modelliert und verarbeitet werden, um ein flexibles Wechseln zwischen Arbeitsschritten und Bearbeitungsreihenfolgen zu ermöglichen, gleichzeitig aber die kausalen, konstruktionsbedingten Zusammenhänge des Arbeitsprozesses zu wahren?



Die Verknüpfung des Arbeitsprozesses und dessen Arbeitsschritte mit Kontextinformationen zu den zu verwendenden Materialien und Werkzeugen, den durchzuführenden Tätigkeiten sowie den relevanten Arbeitsbereichen des Arbeitsplatzes ermöglicht die semantische Beschreibung der Arbeitssituation (vgl. Abschnitt 4.3.3). Einzelne Arbeitsschritte des Arbeitsprozesses können dadurch von Mensch und Assistenzsystem eindeutig klassifiziert und anhand der vorliegenden Arbeitssituation identifiziert werden (vgl. Forschungsfrage 2).

**Forschungsfrage 2:**

Wie kann der Arbeitsprozess semantisch beschrieben werden, um eine einheitliche Interpretation für Mensch und Assistenzsystem und somit die Erkennung des sich in Bearbeitung befindenden Arbeitsschrittes zu ermöglichen?

Zur Sicherstellung, dass die gestellten Forschungsfragen hinreichend bearbeitet und die Zielsetzung dieser Arbeit erreicht wurde, ist eine Umsetzung und Evaluation des entwickelten und modulierten Assistenzsystems erfolgt. Hierzu wurde die Architektur des bestehenden Assistenzsystems der *SmartFactory*<sup>KL</sup> an die Modellierung angepasst und die entwickelte Verarbeitungslogik inklusive Workflow-Modell in das System integriert. Die Evaluation erfolgte anhand der unterstützten Reparatur eines elektrischen Stellantriebes und konnte zeigen, dass das entwickelte System alle Anforderungen hinsichtlich eines kontextsensitiven, modularen Aufbaus erfüllen kann und sich in bestehende Systeme integrieren lässt.

Mit Hinblick auf den aktuellen Stand der Technik erweitert die entwickelte Verarbeitungslogik bisher bekannte Systeme und Ansätze, in denen vornehmlich rein deterministische oder rein probabilistische Prozesse betrachtet wurden, um ein kombiniertes Konzept, das Ansätze beider Prozessbetrachtungen miteinander vereint. Hierdurch können Einschränkungen der jeweiligen Einzelmodelle, wie eine aufwändig Prozessmodellierung oder eines statischen Prozessablaufs, vermindert oder gar vermieden werden. Die Umsetzung und Evaluation bilden den Machbarkeitsnachweis des modellierten Assistenzsystems und der entwickelten Teilkomponenten.

Zusammengefasst bildet diese Arbeit die Grundlage zur praktischen Umsetzung von kognitiven Assistenzsystemen, die sich flexibel an die individuellen Arbeitsweisen von Menschen anpassen und gleichzeitig einen modularen und systemunabhängigen Aufbau ermöglichen. Die entwickelte Systemarchitektur zeigt wie bestehende Assistenzsysteme angepasst werden können. Das semantische Prozessmodell liefert das Rahmenwerk zur Erstellung von dynamischen Arbeitsplänen.

# Anhang

## A Erfassungsbögen für die Informationsakquise und Informationsaufbereitung

### ActiveAssist

#### System

### ActiveAssist

Bosch Rexroth AG

#### Quellen

Literaturrecherche  
Messebesuche  
Forschungsprojekt DAKARA

#### Erfassung (Eingabe)

- Handerkennung
- Ultraschall (Stabschrauber)
- RFID-/Barcode-Leser

Erweiterbar

#### Visualisierung (Ausgabe)

- Monitore
- Projektor
- Pick-to-Light

Erweiterbar



Bildquelle: Bosch Rexroth AG

#### Prozessverarbeitung

Vergleich von eingehenden  
Sensorwerten mit erwarteten  
Werten aus Prozessmodell

#### Prozessmodell

- Deterministisch verkettete Arbeitsschritte
- verschiedene Erfahrungslevel (Aufgaben/Unteraufgaben)

#### Besonderheiten

- Modulares webbasiertes System
- Erweiterbar
- Qualitätskontrolle

Literatur: [@Rexroth 1] [@Rexroth 2]

**cubu:S**

## System

**cubu:S**

Schnaithmann Maschinenbau  
GmbH

## Quellen

Literaturrecherche

## Erfassung (Eingabe)

Multisensorkamera zur

- Bewegungserkennung
- Bauteilerkennung

Erweiterbar

## Visualisierung (Ausgabe)

Projektor für

- Pick-to-Light
- Bild, Video
- Arbeitsanweisung

## Prozessverarbeitung

Vergleich von eingehenden  
Sensorwerten mit erwarteten  
Werten aus Prozessmodell

## Prozessmodell

- Deterministisch verkettete  
Arbeitsschritte

## Besonderheiten

- Basiert auf ulixes „Der  
Assistent“
- Qualitätskontrolle
- Erweiterbar durch Apps

Literatur: [@Ulixes] [@Schnaithmann]

## Der Assistent

### System

# Der Assistent

Ulixes Robotersysteme GmbH

### Quellen

Literaturrecherche

### Erfassung (Eingabe)

Multisensorkamera zur

- Bewegungserkennung
- Bauteilerkennung

### Visualisierung (Ausgabe)

Projektor für

- Pick-to-Light
- Bild, Video
- Arbeitsanweisung

### Prozessverarbeitung

Vergleich von eingehenden  
Sensorwerten mit erwarteten  
Werten aus Prozessmodell

### Prozessmodell

- Deterministisch verkettete  
Arbeitsschritte

### Besonderheiten

- Qualitätskontrolle
- Erweiterbar durch Apps

Literatur: [@Ulixes]

**Der schlaue Klaus**

## System

# Der schlaue Klaus

Optimum GmbH

## Quellen

Literaturrecherche

## Erfassung (Eingabe)

- Kamera und Objektiv

## Visualisierung (Ausgabe)

- HD-Touch-Monitor

## Prozessverarbeitung

Kamerabasierte  
Qualitätskontrolle /  
Bildvergleich als  
Weiterschaltbedingung

## Prozessmodell

- Arbeitsschritte entweder mit
- Deterministischer  
Verkettung oder
  - ohne Verkettung

## Besonderheiten

- Erweiterbar über I/O-  
Module

Literatur: [@OPTIMUM]

## MonSiKo

### System

# MonSiKo

Fraunhofer IPA

### Quellen

Literaturrecherche

### Erfassung (Eingabe)

- Optische 3D-Sensoren (Objekt- und Handposenerkennung)
- Akustische Sensoren
- Spracheingabe

### Visualisierung (Ausgabe)

- Touch-Display



Bildquelle: Fraunhofer IPA

### Prozessverarbeitung

3D-Szenenanalyse zur  
Bestimmung des Arbeitsschrittes

### Prozessmodell

- Variable (probabilistisch)  
verkettete Arbeitsschritte

### Besonderheiten

- Nicht bevormundend
- Fehlererkennung /  
Qualitätskontrolle
- Dokumentation

Literatur: [Fraunhofer] [Ja18] [Wa17]

**motionEAP**

## System

## motionEAP

## Quellen

Literaturrecherche

## Erfassung (Eingabe)

Tiefen und Videokamera zur

- Bewegungserkennung
- Objekterkennung
- Emotionserkennung

## Visualisierung (Ausgabe)

- Projektor

## Prozessverarbeitung

Vergleich von eingehenden  
Sensorwerten mit erwarteten  
Werten aus Prozessmodell

## Prozessmodell

- Deterministisch verkettete Arbeitsschritte
- Verschiedene Erfahrungslevel

(Anfänger, Fortgeschrittener, Experte)

## Besonderheiten

- Gamification zur Motivation
- Qualitätskontrolle
- Direktes Fehlerfeedback

Literatur: [ @motionEAP 2 ] [ Ko16 ] [ Fu16b ]

## Plant@Hand

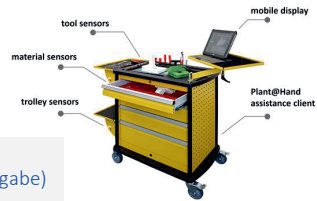
### System

# Plant@Hand

Fraunhofer IGD

### Quellen

Literaturrecherche



Bildquelle: [AB15]

### Erfassung (Eingabe)

- Drucksensoren (Kleinteile)
- Photodiode (Handeingriffe)
- RFID-Reader (Werkzeuge)

### Visualisierung (Ausgabe)

- LCD Touch-Display

### Prozessverarbeitung

- Bestimmung der optimalen sequenziellen Montage-reihenfolge hin zum vorab bekannten Ziel
- Vergleich von digitalem Abbild der Arbeitssituation mit Prozessmodell

### Prozessmodell

- Semantisch verkettete Arbeitsschritte je Komponente
- Semantisch verkettete Komponenten

### Besonderheiten

- Fokussiert visuelle Bereitstellung von Handlungsempfehlungen

Literatur: [Ae16] [AB15]



**ProMiMo**

## System

**ProMiMo**

CoR-Lab Universität Bielefeld  
Spitzencluster it's OWL

## Quellen

Literaturrecherche

## Erfassung (Eingabe)

Kamera zur

- Objekterkennung
- Handerkennung

RFID-Technologie zur

- Auftragserfassung

## Visualisierung (Ausgabe)

- Projektor
- Monitor

## Prozessverarbeitung

Vergleich von eingehenden  
Sensorwerten mit erwarteten  
Werten aus Prozessmodell

## Prozessmodell

- Deterministisch verkettete Arbeitsschritte
- verschiedene Erfahrungslevel (Aufgaben/Unteraufgaben)

## Besonderheiten

- Qualitätskontrolle
- Nutzergetriebene Annotation von Hinweisen und Fehlern

Literatur: [ @OWL ] [ Rü14 ]

## B Dokumente für den Systemtest

### B.1 Erfassungsbogen für die Testdurchläufe

Testgruppe: ☐ Informierte Tester ☐ Unerfahrene Tester ☐ Erfahrene Tester

Testlauf: ☐ 1 ☐ 2 ☐ 3 ☐ 4

#### Proband\*in

Hat das System auf durchgeführte Arbeiten reagiert und eine Unterstützung für den richtigen Arbeitsschritt geliefert? ☐ Ja ☐ Nein

Wenn ja, wie zeitlich passend kam die Unterstützung? ☐ zu früh ☐ zu spät ☐ passend

Als wie gut empfinden Sie die Erkennung des bearbeiteten Arbeitsschrittes? ☐ gut ☐ neutral ☐ schlecht

Wie gut empfinden Sie die Erkennung einer erfolgreichen Bearbeitung des Arbeitsschrittes? ☐ gut ☐ neutral ☐ schlecht

Sollte das System die erfolgreiche Bearbeitung automatisch erkennen? ☐ Ja ☐ Nein

#### Betreuer\*in

Konnte der gesamte Prozess durchgeführt werden? ☐ Ja ☐ Nein

Wie häufig hat das System den Zustand gewechselt? \_\_\_\_\_

Wie häufig durch eine vermeintlich bewusste Entscheidung des Probanden/ der Probandin? \_\_\_\_\_

Wurden die Rahmenbedingungen eingehalten? ☐ Ja ☐ Nein

## B.2 Arbeitsauftrag und Anwendungsszenario

### Beschreibung des Anwendungsszenarios

Das Anwendungsszenario umfasst die Reparatur eines technischen Gerätes (Stellantrieb) an einem Handarbeitsplatz. Zur Unterstützung bei der Aufgabe ist der Handarbeitsplatz mit einem vorbereiteten Arbeitsbereich, einem Assistenzsystem sowie Lagermöglichkeiten für demontierte Teile ausgestattet.

### Beschreibung des Arbeitsplatzes

Der Aufbau des Arbeitsplatzes und dessen Arbeitsbereich sind in Abbildung 1 dargestellt. Das System besteht aus einer Projektionslösung und einem Monitor zur Bereitstellung von Arbeitsanweisungen und Hilfestellungen bei der Durchführung von Arbeitsschritten. Im Bereich des Arbeitsraumes verfügt der Arbeitsplatz über einen zentralen Arbeitsbereich zur Montage / Demontage von Bauteilen sowie einem Lagerbereich für große Teile (links) und für kleinere Teile (rechts). Werkzeuge, die zur Durchführung der Arbeitsschritte benötigt werden, finden sich hängend hinter dem Arbeitsbereich.

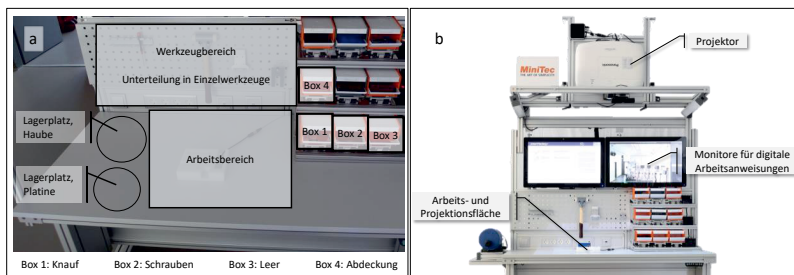


Abbildung 1: Arbeitsplatz mit relevanten Bereichen für den Arbeitsauftrag (a) und Aufbau des Arbeitsplatzes (b)

### Beschreibung des Produktes

Bei dem zu reparierenden technischen System handelt es sich um einen elektrisch gesteuerten Stellantrieb. Dieser besteht aus sechs für dieses Szenario relevanten Bauteilen. Abbildung 2 zeigt die sechs Bauteile mit einer namentlichen Beschriftung. Zur Durchführung einer vollständigen Demontage, wie in der Abbildung gezeigt, müssen neun Arbeitsschritte durchgeführt werden, wobei verschiedene Bearbeitungsreihenfolgen gibt, die lediglich durch den konstruktiven Aufbau des Systems bestimmt werden.

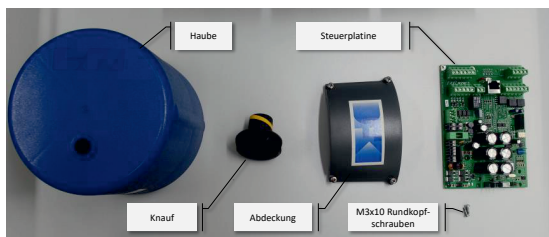


Abbildung 2: Einzelkomponenten des elektrischen Stellantriebes mit Beschriftung

## Rahmenbedingungen

Für den Auftrag benötigte Werkzeuge finden sich an einem Aufbewahrungsort hinter der Arbeitsfläche. Zu jedem Zeitpunkt darf nur ein einziges Werkzeug verwendet werden. Nach Gebrauch des Werkzeuges ist dieses wieder an seinen Aufbewahrungsort zu hängen. Bei der Verwendung eines Werkzeuges ist auf die richtige Auswahl zu achten. Das heißt für „große Schrauben“ wird ein „großer Schraubendreher“ verwendet.

Für demontierte Teile des Stellantriebes sind festgelegte Lagerplätze oder Lagerboxen vorgesehen (siehe Abbildung 1 und „Beschreibung des Arbeitsplatzes“). Nachdem ein Bauteil demontiert wurde ist dieses an den richtigen Lagerplatz abzulegen.

Eine betreuende Person überwacht die Bearbeitung des Arbeitsauftrages und steht bei technischen Problemen des Systems zur Hilfe. Fragen zur Durchführung eines Arbeitsschrittes werden nicht beantwortet.

## Aufgabenbeschreibung

Beim vorliegenden Stellantrieb wurde ein Defekt der Steuerelektronik festgestellt. Ihre Aufgabe ist es alle notwendigen Gehäuseteile zu entfernen, sodass Sie die Steuerplatine lösen und entfernen können. Die entfernten Teile sollen dabei an ihren vorgesehenen Lagerplätzen abgelegt werden. Nachdem die Steuerplatine erfolgreich entfernt wurde, wird der Austausch gegen eine neue Steuerplatine simuliert. Bauen Sie hierzu die alte Platine wieder in den Stellantrieb ein und stellen Sie alle elektrischen Verbindungen her. Anschließend montieren Sie die demontierten Gehäuseteile bis sich der Stellantrieb wieder in seinem Ursprungszustand befindet.

### B.3 Nutzungsanweisung des Assistenzsystem

#### Benutzeroberfläche und Unterstützungsfunktionen

Die Hauptbestandteile des Assistenzsystems sind eine Benutzeroberfläche, eine digitale Arbeitsanweisung sowie eine In-Situ-Projektion (vgl. Abbildung 1a). Die Benutzeroberfläche findet sich auf dem linken Monitor und dient im Rahmen der Versuchsdurchführung lediglich zum Starten der Unterstützung für den Arbeitsauftrag (vgl. Abbildung 1b).

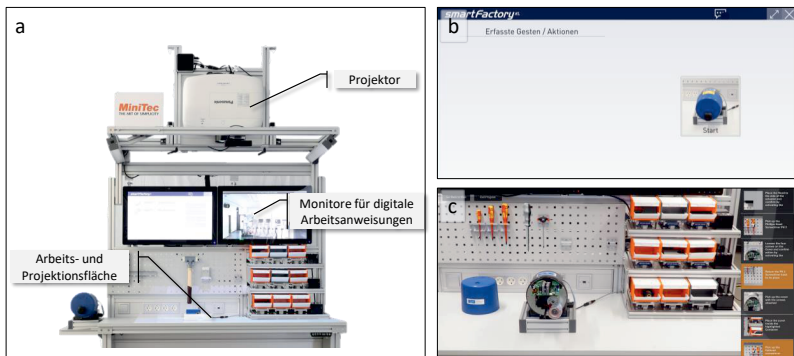


Abbildung 1: Bestandteile des Assistenzsystems (a), Benutzeroberfläche des Systems (b) und digitale Arbeitsanweisung (c)

Die digitale Arbeitsanweisung befindet sich auf dem rechten Monitor und zeigt detailliertere Informationen zur Durchführung des aktuellen Arbeitsschrittes. Hierzu gehören textuelle Beschreibungen am unteren Rand des Displays sowie ein Abbild der Arbeitssituation mit überlagerten Animationen (vgl. Abbildung 1c). Im Falle von komplexen Arbeitsanweisungen mit mehreren Teilschritten schaltet das System zyklisch durch alle Schritte durch. Zum Beispiel für den Fall: 1) Werkzeug nehmen, 2) Werkzeug benutzen, 3) Werkzeug zurücklegen.

Die In-Situ-Projektion stellt zusätzlich zur digitalen Arbeitsanweisung eine weitere Unterstützungshilfe in Form von projizierten Symbolen bereit. Hierzu gehören farbige Rechtecke zur Hervorhebung von Werkzeugen, Lagerboxen oder Bauteilen sowie die Visualisierung eines digitalen Buttons zur Bestätigung von Arbeitsschritten (vgl. Abbildung 2).

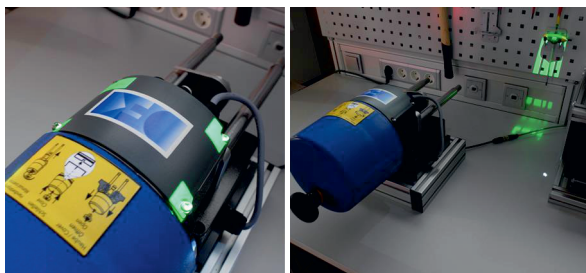


Abbildung 2: Unterstützungshilfen durch die In-Situ-Projektion während eines Arbeitsschrittes

### Direkte Interaktionen

Eine direkte Interaktion mit dem Assistenzsystem ist lediglich für den Startvorgang und die Bestätigung einer erfolgreichen Bearbeitung eines Arbeitsschrittes vorgesehen. Zum Starten des Vorgangs wird der Button mit dem Bild des Stellantriebes auf dem linken Monitor verwendet (vgl. Abbildung 1b). Zur Bestätigung eines erfolgreichen Arbeitsschrittabschlusses steht ein virtueller Button als Projektion auf die Arbeitsplatte bereit (vgl. Abbildung 2). Eine direkte Auswahl des Arbeitsschrittes, der bearbeitet werden soll, steht nicht zur Verfügung.

### Indirekte Interaktionen

Durch die Bearbeitung des Arbeitsauftrages und die erfolgreiche Bearbeitung von Arbeitsschritten ist eine indirekte Interaktion mit dem Assistenzsystem möglich. Hierzu erfasst das System die Arbeitssituation am Handarbeitsplatz und bietet Unterstützung bei der Durchführung eines Arbeitsschrittes. Nach erfolgreicher Bearbeitung des Arbeitsauftrages stoppt das System automatisch.

## B.4 Systembeschreibung zur Funktionsweise des Assistenzsystems und der Verarbeitungslogik

### Grundlegende Funktionsweise

Die Arbeitsweise des Assistenzsystems basiert auf zwei Grundprinzipien, durch die bestimmt werden wann und zu welchem Arbeitsschritt eine Unterstützung angezeigt wird.

Grundprinzip Nummer eins ist die Annahme, dass es verschiedene Bearbeitungsreihenfolgen für einen Arbeitsauftrag geben kann und dass die bearbeitende Person den Arbeitsweg selbst bestimmen kann.

(Nähere Erläuterungen zur Umsetzung in „Einschränkung der Arbeitsschritte für die eine Unterstützung geboten wird“)

Grundprinzip Nummer zwei ist die Annahme, dass durch die Beobachtungen der bearbeitenden Person und durch das Wissen, welche Arbeitsschritte gerade durchgeführt werden können, ermittelt werden kann, für welchen Arbeitsschritt eine Hilfestellung notwendig sein könnte.

(Nähere Erläuterungen zur Umsetzung in „Bestimmung des Arbeitsschrittes für den eine Unterstützung angezeigt wird“)

Durch die Kombination beider Grundprinzipien kann das Assistenzsystem bestimmen, welche Arbeitsschritte tatsächlich durchführbar sind und für welchen dieser Schritte eine Hilfestellung angezeigt wird, in Abhängigkeit von den Tätigkeiten der bearbeitenden Person.

### Einschränkung der Arbeitsschritte für die eine Unterstützung geboten wird

Zur Bestimmung welche Arbeitsschritte zum aktuellen Zeitpunkt durchgeführt werden können, wird eine Modellierung aller Arbeitsschritte des Prozesses und deren Montagereihenfolgen durchgeführt. Abbildung 1 zeigt den Ausschnitt einer solchen Modellierung für fünf Arbeitsschritte ( $s_2 - s_6$ ), sowie den Verlauf, der als durchführbar angesehenen Arbeitsschritte. Die möglichen Bearbeitungsreihenfolgen ergeben sich durch die Verkettung der Schritte mittels Pfeilen. Dabei gilt ein Arbeitsschritt genau dann als durchführbar, wenn alle direkt vorangehenden Arbeitsschritte bearbeitet wurden (in der Abbildung grün hinterlegt). Für Abbildung 1a bedeutet das, dass Schritt  $s_2$  erfolgreich bearbeitet wurde und dass Schritt  $s_3$  und  $s_4$  bearbeitet werden können. In Abbildung 1b wurde Schritt  $s_3$  bearbeitet, sodass nur noch Schritt  $s_4$  bearbeitet werden kann. Zur Bearbeitung der Schritte  $s_5$  und  $s_6$  fehlt die erfolgreiche Bearbeitung des Schrittes  $s_4$ . Abbildung 1c zeigt den Zustand nach zusätzlicher Bearbeitung des Schrittes  $s_4$ .

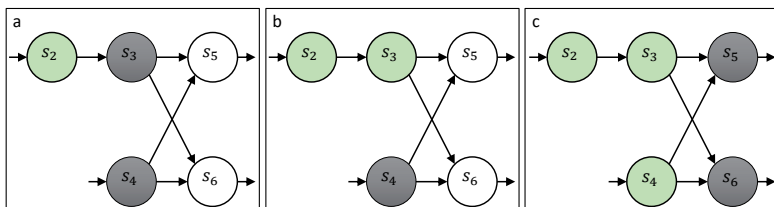


Abbildung 1: Verkettung von Arbeitsschritten und Visualisierung der bearbeitbaren Arbeitsschritte (grau) im Verlauf des Arbeitsprozesses.

### Bestimmung des Arbeitsschrittes für den eine Unterstützung angezeigt wird

Die Auswahl des durchführbaren Arbeitsschrittes, für den eine Unterstützung geboten wird, basiert auf der Erfassung der folgenden vier Kontextinformationen mit Bezug zur Arbeitssituation (vgl. Abbildung 2a):

1. Welches Werkzeug wird verwendet
2. In welchem Bereich des Arbeitsplatzes agiert die arbeitende Person
3. Welche Bauteile / Komponenten werden verwendet
4. Welche Tätigkeit / Aktivität führt die bearbeitende Person durch

Die Erfassung der Kontextinformationen erfolgt über eine Tiefenbildkamera mit entsprechender Auswertung des Tiefenbildes. Für kleinere Komponenten / Bauteile (3) erfolgt zudem eine Erfassung über ein Wägezellensystem, dass die Entnahme und Ablage von Bauteilen registriert.

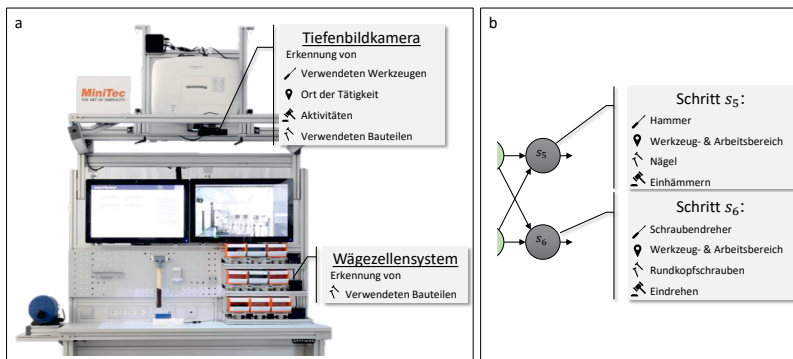


Abbildung 2: Erfassung von Kontextinformationen (a) und Verknüpfung von Kontextinformationen mit Arbeitsschritten (b)

Um eine Auswahl des Arbeitsschrittes auf Basis dieser Informationen durchzuführen, wird darüber hinaus jeder Arbeitsschritt mit den Kontextinformationen annotiert, die zur Bearbeitung dieses Schrittes relevant sind (vgl. Abbildung 2b). Durch einen Vergleich der zuletzt erfassten Kontextinformationen mit den Annotationen der Arbeitsschritte, die durchgeführt werden können, berechnet das System den Grad der Übereinstimmung. Für den Arbeitsschritt mit der größten Übereinstimmung wird eine Hilfeleistung angezeigt.

Für das dargestellte Beispiel bedeutet das, dass bei einer Erkennung von „Hammer“, „Werkzeug- & Arbeitsbereich“, „Nägel“ und „Eindrehen“ der Arbeitsschritt  $s_5$  angezeigt wird (75% Übereinstimmung  $s_5$  vs. 50% Übereinstimmung  $s_6$ ).

Ändert sich die erfasste Kontextinformation „Hammer“ zu „Schraubendreher“ wird eine Unterstützung für Schritt  $s_6$  angezeigt (75% Übereinstimmung  $s_6$  vs. 50% Übereinstimmung  $s_5$ )



# Literatur

## Monografien und Artikel

- [AAU15] Alm, R.; Aehnelt, M.; Urban, B.: Plant@hand: From activity recognition to situation-based annotation management at mobile assembly workplaces. In: Proceedings of the 2nd international Workshop on Sensor-based Activity Recognition and Interaction. S. 15, 2015.
- [Ab13] Abel, D.: Petri-Netze für Ingenieure: Modellbildung und Analyse diskret gesteuerter Systeme. Springer-Verlag, 2013.
- [AB15] Aehnelt, M.; Bader, S.: From information assistance to cognitive automation: A smart assembly use case. In: International Conference on Agents and Artificial Intelligence. S. 207–222, 2015.
- [AB16] Aehnelt, M.; Bader, S.: Providing and Adapting Information Assistance for Smart Assembly Stations. In: Proceedings of SAI Intelligent Systems Conference. S. 540–562, 2016.
- [Ab17] Abramovici, M.; Wolf, M.; Adwernat, S.; Neges, M.: Context-aware maintenance support for augmented reality assistance and synchronous multi-user collaboration. Procedia CIRP 59, S. 18–22, 2017.
- [Ab99] Abowd, G. D.; Dey, A. K.; Brown, P. J.; Davies, N.; Smith, M.; Steggles, P.: Towards a better understanding of context and context-awareness. In: International symposium on handheld and ubiquitous computing. S. 304–307, 1999.
- [AGU14] Aehnelt, M.; Gutzeit, E.; Urban, B.: Using activity recognition for the tracking of assembly processes: Challenges and requirements. WOAR 2014, S. 12–21, 2014.
- [Ah13] Ahrens, W.: Eine Gegenüberstellung von VDI/VDE 3682, PROLIST, eCI@ss. atp magazin 52/09, S. 32–45, 2013.
- [AH15] Alm, R.; Hadlak, S.: Towards integration and management of contextualized information in the manufacturing environment by digital annotations. In: SUMMER SCHOOL. Bd. 141, 2015.

- [Al15] Allweyer, T.: BPMN 2.0-Business Process Model and Notation: Einführung in den Standard für die Geschäftsprozessmodellierung. BoD–Books on Demand, 2015.
- [AM16] Aehnel, M.; Müller, A.: Werker der Zukunft: Assistenz im Zeitalter von Industrie 4.0. In: VDE-Kongress 2016. VDE Verlag, Berlin, 2016, ISBN: 9783800743087.
- [Ap18] Apt, W.; Bovenschulte, M.; Priesack, K.; Weiß, C.; Hartmann, E. A.: Einsatz von digitalen Assistenzsystemen im Betrieb. Forschungsbericht vom Institut für Innovation und Technik im Auftrag des BMAS, 2018.
- [AR11] Abele, E.; Reinhart, G.: Zukunft der Produktion. Carl Hanser Verlag GmbH & Co. KG, München, 2011, ISBN: 978-3-446-42595-8.
- [Ba08] Bannat, A.; Wallhoff, F.; Rigoll, G.; Friesdorf, F.; Bubb, H.; Stork, S.; Müller, H. J.; Schubö, A.; Wiesbeck, M.; Zäh, M. F. et al.: Towards optimal worker assistance: A framework for adaptive selection and presentation of assembly instructions. In: Proceedings of the 1st international workshop on cognition for technical systems, Cotesys. 2008.
- [BA14] Bader, S.; Aehnel, M.: Tracking Assembly Processes and Providing Assistance in Smart Factories. In: ICAART (1). S. 161–168, 2014.
- [Bä15] Bächler, L.; Bächler, A.; Kölz, M.; Hörz, T.; Heidenreich, T.: Über die Entwicklung eines prozedural-interaktiven Assistenzsystems für leistungsgeminderte und-gewandelte Mitarbeiter in der manuellen Montage. 2015.
- [Ba17] Bauernhansl, T.: Die vierte industrielle Revolution – Der Weg in ein wertschaffendes Produktionsparadigma. In: Handbuch Industrie 4.0 Bd. 4. Springer, S. 1–31, 2017.
- [Ba96a] Baumgarten, B.: Petri-Netze: Grundlagen und Anwendungen. Spektrum, Akademischer Verlag, 1996.
- [Ba96b] Baumöl, U.; Borchers, J.; Eicker, S.; Hildebrand, K.; Jung, R.; Lehner, F.: Einordnung und Terminologie des Software Reengineering. Informatik-Spektrum 19/4, S. 191–195, 1996, issn: 0170-6012.
- [BB16] Bahubalendruni, M. R.; Biswal, B. B.: A review on assembly sequence generation and its automation. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 230/5, S. 824–838, 2016.
- [Be17] Bertram, P.; Birtel, M.; Quint, F.; Ruskowski, M.: Informationsmodellierung zur Beschreibung manueller Tätigkeiten an Handarbeitsplätzen. Mensch und Computer 2017-Workshopband, 2017.

- [Be18a] Becker, T.: Prozesse in Produktion und Supply Chain optimieren. Springer-Verlag Berlin, 2018.
- [Be18b] Bertram, P.; Birtel, M.; Quint, F.; Ruskowski, M.: Intelligent Manual Working Station through Assistive Systems. IFAC-PapersOnLine 51/11, S. 170–175, 2018.
- [Be19b] Bertram, P.; Motsch, W.; Rübel, P.; Ruskowski, M.: Intelligent Material Supply Supporting Assistive Systems for Manual Working Stations. 29th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM), 2019.
- [Bi18] Birtel, M.; Mohr, F.; Hermann, J.; Bertram, P.; Ruskowski, M.: Requirements for a Human-Centered Condition Monitoring in Modular Production Environments. IFAC-PapersOnLine 51/11, S. 909–914, 2018.
- [BMR19] Birtel, M.; Motsch, W.; Ruskowski, M.: Implementierung eines mitarbeiterrollenbasierten Informationssystems in einer modularen Produktionsumgebung mittels einer menschenzentrierten Verwaltungsschale. Mensch und Computer 2019-Workshopband, 2019.
- [Bo17] Bosch, T.; Könemann, R.; de Cock, H.; van Rhijn, G.: The effects of projected versus display instructions on productivity, quality and workload in a simulated assembly task. In: Proceedings of the 10th International Conference on Pervasive Technologies Related to Assistive Environments. S. 412–415, 2017.
- [BPV12] Becker, J.; Probandt, W.; Vering, O.: Modellierungssprachen. In: Grundsätze ordnungsmäßiger Modellierung. Springer, S. 4–30, 2012.
- [BQR17] Birtel, M.; Quint, F.; Ruskowski, M.: Virtuelle Benutzungsschnittstellen auf Basis semantischer Modelle zur vereinfachten Anlageninteraktion, 2017.
- [Br11] Broy, M.: Cyber-physical systems: Innovation durch softwareintensive eingebettete Systeme. Springer-Verlag, 2011.
- [Br19] Brandau, C.: Modellierung und Transformation digitaler Schaltungen mittels Digital Circuit Petri Nets. Springer, 2019.
- [Br95] Brown, P. J.: The stick-e document: A framework for creating context-aware applications. Electronic Publishing-Chichester- 8, S. 259–272, 1995.
- [Ch00] Cheverst, K.; Davies, N.; Mitchell, K.; Friday, A.; Efstratiou, C.: Developing a context-aware electronic tourist guide: Some issues and experiences. In: Proceedings of the SIGCHI conference on Human Factors in Computing Systems. S. 17–24, 2000.

- [CH02] Cook, A. M.; Hussey, S. M.: Assistive technologies: Principles and practice. St. Louis: Mosby-Year Book, 2002.
- [Cl18] Claeys, A.; Hoedt, S.; Schamp, M.; van Landeghem, H.; Cottyn, J.: Ontological Model for Managing Context-Aware Assembly Instructions. IFAC-PapersOnLine 51/11, S. 176–181, 2018.
- [Cr02] Crowley, J. L.; Coutaz, J.; Rey, G.; Reignier, P.: Perceptual components for context aware computing. In: International conference on ubiquitous computing. S. 117–134, 2002.
- [Cs75] Csikszentmihalyi, M.: Beyond boredom and anxiety. San Francisco, CA, US: Jossey-Bass, 1975.
- [De01] Dey, A. K.: Understanding and using context. Personal and ubiquitous computing 5/1, S. 4–7, 2001.
- [De09] Deacon, J.: Model-view-controller (mvc) architecture. 2009, URL: <http://www.jdl.co.uk/briefings/MVC.pdf>.
- [Dw18] Dworatschek, S.: Einführung in die Datenverarbeitung. De Gruyter, Berlin, 2018, ISBN: 3111683028.
- [Ed96] Eddy, S. R.: Hidden markov models. Current opinion in structural biology 6/3, S. 361–365, 1996.
- [Eu03] Eugster, P. T.; Felber, P. A.; Guerraoui, R.; Kermarrec, A.-M.: The many faces of publish/subscribe. ACM Computing Surveys 35/2, S. 114–131, 2003, ISSN: 03600300.
- [Fe17] Ferrer, B. R.; Mohammed, W. M.; Lobov, A.; Galera, A. M.; Lastra, J. L. M.: Including human tasks as semantic resources in manufacturing ontology models. In: IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society. S. 3466–3473, 2017.
- [FKS16] Funk, M.; Kosch, T.; Schmidt, A.: Interactive worker assistance: Comparing the effects of in-situ projection, head-mounted displays, tablet, and paper instructions. In: Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. S. 934–939, 2016.
- [Fo98] Fosler-Lussier, E.: Markov Models and Hidden Markov Models: A Brief Tutorial. International Computer Science Institute, 1998, URL: <http://di.ubi.pt/~jpaulo/competence/tutorials/hmm-tutorial-1.pdf>, Stand: 26. 12. 2019.
- [FS15] Funk, M.; Schmidt, A.: Cognitive assistance in the workplace. IEEE Pervasive Computing 14/3, S. 53–55, 2015.

- [Fu15] Funk, M.; Bächler, A.; Bächler, L.; Korn, O.; Krieger, C.; Heidenreich, T.; Schmidt, A.: Comparing projected in-situ feedback at the manual assembly workplace with impaired workers. In (Makedon, F., Hrsg.): Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments. ACM, New York, NY, S. 1–8, 2015, ISBN: 9781450334525.
- [Fu16b] Funk, M.; Kosch, T.; Kettner, R.; Korn, O.; Schmidt, A.: motionEAP: An overview of 4 years of combining industrial assembly with augmented reality for industry 4.0. In: Proceedings of the 16th international conference on knowledge technologies and datadriven business. S. 4, 2016.
- [Ga17] Gadatsch, A.: Grundkurs Geschäftsprozess-Management: Analyse, Modellierung, Optimierung und Controlling von Prozessen. Springer-Verlag, 2017.
- [GB12] Geisberger, E.; Broy, M.: agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems. Springer-Verlag, 2012.
- [GC96] Gram, C.; Cockton, G.: Design Principles for Interactive Software. Springer US, Boston, MA, 1996, ISBN: 978-1-4757-4944-1.
- [Go16] Gorecky, D.; Weyer, S.; Hennecke, A.; Zühlke, D.: Design and instantiation of a modular system architecture for smart factories. IFAC-PapersOnLine 49/31, S. 79–84, 2016.
- [Gr01] Grzega, J.: Sprachwissenschaft ohne Fachchinesisch: 7 aktuelle Studien für alle Sprachinteressierten. Shaker, Aachen, 2001, ISBN: 3-8265-8826-6.
- [Hä04] Häuslein, A.: Systemanalyse: Grundlagen, Techniken, Notierungen. VDE Verlag, 2004.
- [Ha14] Ha, K.; Chen, Z.; Hu, W.; Richter, W.; Pillai, P.; Satyanarayanan, M.: Towards wearable cognitive assistance. In: Proceedings of the 12th annual international conference on Mobile systems, applications, and services. S. 68–81, 2014.
- [Ha18] Hadj, R. B.; Belhadj, I.; Trigui, M.; Aifaoui, N.: Assembly sequences plan generation using features simplification. Advances in Engineering Software 119, S. 1–11, 2018.
- [Ha19] Haberfellner, R.; de Weck, O.; Fricke, E.; Vössner, S.: Systems Engineering. Springer International Publishing, Cham, 2019, ISBN: 978-3-030-13430-3.
- [Ha87] Harel, D.: Statecharts: A visual formalism for complex systems. Science of computer programming 8/3, S. 231–274, 1987.

- [HB14] Hadar, R.; Bilberg, A.: Reconfigurable Manufacturing System Design and Implementation-An Industrial Application at a Manufacturer of Consumer Goods. In: Enabling Manufacturing Competitiveness and Economic Sustainability. Springer, S. 455–460, 2014.
- [HB19] Hinrichsen, S.; Bornewasser, M.: How to Design Assembly Assistance Systems. In (Karwowski, W.; Ahram, T., Hrsg.): Intelligent Human Systems Integration 2019. Springer International Publishing, Cham, S. 286–292, 2019, ISBN: 978-3-030-11051-2.
- [HLS12] Hodek, S.; Loskyll, M.; Schlick, J.: Semantische Informationsmodelle zur Feldgeräteansteuerung - Plug&Pay-Geräteintegration und dynamische Orchestrierung von Produktionsprozessen. In: atp edition - Automatisierungstechnische Praxis. Bd. 54, DIV Deutscher Industrieverband, S. 44–51, 2012.
- [HSB11] Hompel, M.; Sadowsky, V.; Beck, M.: Kommissionierung: Materialflusssysteme 2-Planung und Berechnung der Kommissionierung in der Logistik. Springer-Verlag, 2011.
- [HUM02] Hopcroft, J. E.; Ullman, J. D.; Motwani, R.: Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie. Pearson Studium Deutschland, München, 2002.
- [Ja09] Jacobsen, H.-A.: Publish/Subscribe. In (LIU, L.; ÖZSU, M. T., Hrsg.): Encyclopedia of Database Systems. Springer US, Boston, MA, S. 2208–2211, 2009, ISBN: 978-0-387-39940-9, URL: [https://doi.org/10.1007/978-0-387-39940-9\\_1181](https://doi.org/10.1007/978-0-387-39940-9_1181).
- [JR91] Juang, B. H.; Rabiner, L. R.: Hidden Markov models for speech recognition. Technometrics 33/3, S. 251–272, 1991.
- [JV13] Jessen, E.; Valk, R.: Rechensysteme: Grundlagen der Modellbildung. Springer-Verlag, Berlin, 2013.
- [Ka14] Karlin, S.: A first course in stochastic processes. Academic press, New York, 2014.
- [KC05] Kofod-Petersen, A.; Cassens, J.: Using activity theory to model context awareness. In: International Workshop on Modeling and Retrieval of Context. S. 1–17, 2005.
- [Ke11] Kecher, C.: UML 2.0: Das umfassende Handbuch. Galileo Press, 2011.
- [Ke99] Keller, G.: SAP R/3 prozessorientiert anwenden: Iteratives Prozess-Prototyping mit ereignisgesteuerten Prozessketten und Knowledge Maps. Addison-Wesley, 1999.

- [KFS15] Korn, O.; Funk, M.; Schmidt, A.: Assistive systems for the workplace: Towards context-aware assistance. In: Gamification: Concepts, Methodologies, Tools, and Applications. IGI Global, S. 1936–1949, 2015.
- [KN09] Koren, Y.; Ni, J.: Personalized production paradigm. A keynote paper presented at the European Union Manufuture Summit, Gothenburg, Sweden, 2009.
- [Kn17] Knoch, S.; Ponpathirkootam, S.; Fettke, P.; Loos, P.: Technology-enhanced process elicitation of worker activities in manufacturing. In: International Conference on Business Process Management. S. 273–284, 2017.
- [Kn18] Knoch, S.; Herbig, N.; Ponpathirkootam, S.; Kosmalla, F.; Staudt, P.; Fettke, P.; Loos, P.: Enhancing Process Data in Manual Assembly Workflows. In: International Conference on Business Process Management. S. 269–280, 2018.
- [KNS92] Keller, G.; Nüttgens, M.; Scheer, A.-W.: Semantische Prozessmodellierung auf der Grundlage „Ereignisgesteuerter Prozessketten (EPK)“. Institut für Wirtschaftsinformatik (IWi), Universität des Saarlandes, 1992.
- [Ko13] Kokkalis, N.; Köhn, T.; Huebner, J.; Lee, M.; Schulze, F.; Klemmer, S. R.: Taskgenies: Automatically providing action plans helps people complete tasks. ACM Transactions on Computer-Human Interaction (TOCHI) 20/5, S. 27, 2013.
- [Ko14b] Korn, O.; Funk, M.; Abele, S.; Hörz, T.; Schmidt, A.: Context-aware assistive systems at the workplace: Analyzing the effects of projection and gamification. In: Proceedings of the 7th international conference on pervasive technologies related to assistive environments. S. 38, 2014.
- [Ko16] Kosch, T.; Kettner, R.; Funk, M.; Schmidt, A.: MotionEAP—Ein System zur Effizienzsteigerung und Assistenz bei Produktionsprozessen in Unternehmen auf Basis von Bewegungserkennung und Projektion. 2016.
- [KRS10] Koch, I.; Reisig, W.; Schreiber, F.: Modeling in systems biology: The Petri net approach. Springer Science & Business Media, 2010.
- [KSH13] Korn, O.; Schmidt, A.; Hörz, T.: The potentials of in-situ-projection for augmented workplaces in production: A study with impaired persons. In: CHI’13 Extended Abstracts on Human Factors in Computing Systems. S. 979–984, 2013, URL: [http://publica.fraunhofer.de/eprints/urn\\_nbn\\_de\\_0011-n-5238969.pdf](http://publica.fraunhofer.de/eprints/urn_nbn_de_0011-n-5238969.pdf).

- [KWH13] Kagermann, H.; Wahlster, W.; Helbig, J.: Securing the future of German manufacturing industry. Recommendations for implementing the strategic initiative INDUSTRIE 4.0, 2013.
- [Lo12] Loskyll, M.; Heck, I.; Schlick, J.; Schwarz, M.: Context-based orchestration for control of resource-efficient manufacturing processes. *Future Internet* 4/3, S. 737–761, 2012.
- [LW13] Lotter, B.; Wiendahl, H.-P.: Montage in der industriellen Produktion: Ein Handbuch für die Praxis. Springer-Verlag, 2013.
- [LZB17] Long, F.; Zeiler, P.; Bertsche, B.: Modelling the flexibility of production systems in Industry 4.0 for analysing their productivity and availability with high-level Petri nets. *IFAC-PapersOnLine* 50/1, S. 5680–5687, 2017.
- [Ma84] Mayr, E. W.: An algorithm for the general Petri net reachability problem. *SIAM Journal on computing* 13/3, S. 441–460, 1984.
- [Mi90] Mitchell, R.: Managing complexity in software engineering. Peregrinus on behalf of the Institution of Electrical Engineers, Hitchin, 1990, ISBN: 0863411711.
- [Mü16] Müller, R.; Hörauf, L.; Vette, M.; Speicher, C.: Planning and developing cyber-physical assembly systems by connecting virtual and real worlds. *Procedia CIRP* 52, S. 35–40, 2016.
- [Mü18a] Müller, R.; Vette, M.; Hörauf, L.; Speicher, C.; Bashir, A.: Intelligent and Flexible Worker Assistance Systems-Assembly Assistance Platform for Planning Assisted Assembly and Rework as Well as Execution of a Worker-Centered Assistance. In: *VISIGRAPP (2: HUCAPP)*. S. 77–85, 2018.
- [Mü18b] Müller, R.; Vette-Steinkamp, M.; Hörauf, L.; Speicher, C.; Bashir, A.: Worker centered cognitive assistance for dynamically created repairing jobs in rework area. *Procedia CIRP* 72, S. 141–146, 2018.
- [Mu89] Murata, T.: Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* 77/4, S. 541–580, 1989.
- [NR02] Nüttgens, M.; Rump, F. J.: Syntax und Semantik Ereignisgesteuerter Prozessketten (EPK). In: *Promise*. Bd. 2, S. 64–77, 2002.
- [OP17] Orciuli, F.; Parente, M.: An ontology-driven context-aware recommender system for indoor shopping based on cellular automata. *Journal of Ambient Intelligence and Humanized Computing* 8/6, S. 937–955, 2017.
- [Pa98] Pascoe, J.: Adding generic contextual capabilities to wearable computers. In: *2nd international symposium on wearable computers*. S. 92–99, 1998.



- [Pf15] Pfrommer, J.; Stogl, D.; Aleksandrov, K.; Navarro, S. E.; Hein, B.; Beyerer, J.: Plug & produce by modelling skills and service-oriented orchestration of reconfigurable manufacturing systems. *at-Automatisierungstechnik* 63/10, S. 790–800, 2015.
- [Qu16] Quint, F.; Loch, F.; Orfgen, M.; Zuehlke, D.: A System Architecture for Assistance in Manual Tasks. In: *Intelligent Environments (Workshops)*. S. 43–52, 2016.
- [Ra89] Rabiner, L. R.: A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77/2, S. 257–286, 1989.
- [Re10] Reisig, W.: *Petrinetze: Modellierungstechnik, Analysemethoden, Fallstudien*. Springer-Verlag, Wiesbaden, 2010.
- [RH96] Rosenberg, L. H.; Hyatt, L. E.: *Software re-engineering*. Software Assurance Technology Center, S. 2–3, 1996.
- [RJ19] Root, M.; Jauch, C.: Challenges of designing hand recognition for a manual assembly assistance system. In: *Multimodal Sensing: Technologies and Applications*. Bd. 11059, 2019.
- [RJ86] Rabiner, L. R.; Juang, B.-H.: An introduction to hidden Markov models. *IEEE ASSP Magazine* 3/1, S. 4–16, 1986.
- [Ro98] Rodden, T.; Cheverst, K.; Davies, K.; Dix, A.: Exploiting context in HCI design for mobile systems. In: *Workshop on human computer interaction with mobile devices*. S. 21–22, 1998.
- [RR12] Robertson, S.; Robertson, J.: *Mastering the requirements process: Getting requirements right*. Addison-Wesley, 2012.
- [Sc01] Scheer, A.-W.: *ARIS - Modellierungsmethoden, Metamodelle, Anwendungen*. Springer-Verlag, Berlin, 2001, ISBN: 978-3-642-97731-2.
- [Sc10] Schatten, A.; Biffi, S.; Demolsky, M.; Gostischa-Franta, E.; Östreicher, T.; Winkler, D.: *Best practice software-engineering: Eine praxiserprobte Zusammenstellung von komponentenorientierten Konzepten, Methoden und Werkzeugen*. Springer-Verlag, 2010.
- [Sc98] Scheer, A.-W.: *ARIS - vom Geschäftsprozess zum Anwendungssystem*. Springer-Verlag, Berlin, 1998.
- [Se12] Seidl, M.; Brandsteidl, M.; Huemer, C.; Kappel, G.: *UML@ Classroom: Eine Einführung in die objektorientierte Modellierung*. dpunkt.verlag, 2012.

- [SG15] Senderek, R.; Geisler, K.: Assistenzsysteme zur Lernunterstützung in der Industrie 4.0. In: Proceedings der Pre-Conference Workshops der 13. E-Learning Fachtagung Informatik. 2015.
- [Si12] Sipser, M.: Introduction to the Theory of Computation. Cengage learning, 2012.
- [Si19] Singh, A.; Quint, F.; Bertram, P.; Ruskowski, M.: A Framework for Semantic Description and Interoperability across Cyber-Physical Systems. In: International Journal on Advances in Intelligent Systems. Bd. 12, S. 70–81, 2019.
- [Sm04] Smith, S. S.-F.: Using multiple genetic operators to reduce premature convergence in genetic assembly planning. Computers in Industry 54/1, S. 35–49, 2004.
- [Sp13] Spath, D., Hrsg.: Produktionsarbeit der Zukunft - Industrie 4.0. Fraunhofer-Verlag, Stuttgart, 2013, ISBN: 3839605709.
- [St90] Starke, P. H.: Analyse von Petri-netz-modellen. Springer, 1990.
- [ST94] Schilit, B. N.; Theimer, M. M.: Disseminating active map information to mobile hosts. IEEE network 8/5, S. 22–32, 1994.
- [SWD10] Spath, D.; Weisbecker, A.; Drawehn, J.: Business Process Modeling 2010. Modellierung von ausführbaren Geschäftsprozessen mit der Business Process Modeling Notation, Stuttgart, 2010.
- [SWM14] Schenk, M.; Wirth, S.; Müller, E.: Fabrikplanung und Fabrikbetrieb: Methoden für die wandlungsfähige, vernetzte und ressourceneffiziente Fabrik. Springer Vieweg, Berlin, 2014, ISBN: 9783642054587.
- [TH17] Tao, S.; Hu, M.: A contact relation analysis approach to assembly sequence planning for assembly models. Computer-Aided Design and Applications 14/6, S. 720–733, 2017.
- [TS02] Thiemermann, S.; Schulz, O.: Trennung aufgehoben. Computer & Automation 2002/8, S. 82–85, 2002.
- [TT10] Tamm, G.; Tribowski, C.: RFID. Springer, Heidelberg, 2010, ISBN: 9783642114595.
- [TTV09] Terkaj, W.; Tolio, T.; Valente, A.: Focused flexibility in production systems. In: Changeable and reconfigurable manufacturing systems. Springer, S. 47–66, 2009.
- [Va17] Varga, E.: Reengineering and Reverse Engineering. In: Unraveling Software Maintenance and Evolution. Springer, S. 247–289, 2017.

- [Va79] Valette, R.: Analysis of Petri nets by stepwise refinements. *Journal of computer and system sciences* 18/1, S. 35–46, 1979.
- [VS11] Van der Aalst, W.; Stahl, C.: *Modeling business processes: A petri net-oriented approach*. MIT press, 2011.
- [Wa06] Wagner, F.; Schmuki, R.; Wagner, T.; Wolstenholme, P.: *Modeling software with finite state machines: A practical approach*. Auerbach Publications, 2006.
- [Wa14] Wahlster, W.: Semantic technologies for mass customization. In: *Towards the Internet of Services: The THESEUS Research Program*. Springer, S. 3–13, 2014.
- [We15] Weyer, S.; Schmitt, M.; Ohmer, M.; Gorecky, D.: Towards Industry 4.0-Standardization as the crucial challenge for highly modular, multi-vendor production systems. *IFAC-PapersOnLine* 48/3, S. 579–584, 2015.
- [We17b] Weyer, S.; Quint, F.; Fischer, S.; Gorecky, D.; Zühlke, D.: Die SmartFactory für individualisierte Kleinserienfertigung. *Handbuch Industrie 4*, S. 691–708, 2017.
- [Wh04] White, S. A.: Introduction to BPMN. 2004, 2004, URL: [https://www.omg.org/bpmn/Documents/Introduction\\_to\\_BPMN.pdf](https://www.omg.org/bpmn/Documents/Introduction_to_BPMN.pdf), Stand: 26. 10. 2019.
- [Wi14] Wiesbeck, M.: *Struktur zur Repräsentation von Montagesequenzen für die situationsorientierte Werkerführung*. Herbert Utz Verlag, 2014.
- [WJH97] Ward, A.; Jones, A.; Hopper, A.: A new location technique for the active office. *IEEE Personal communications* 4/5, S. 42–47, 1997.
- [WLH10] Wünsch, D.; Lüder, A.; Heinze, M.: Flexibility and re-configurability in manufacturing by means of distributed automation systems—an overview. In: *Distributed Manufacturing*. Springer, S. 51–70, 2010.
- [WT17] Wiesner, S.; Thoben, K.-D.: Cyber-physical product-service systems. In: *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*. Springer, S. 63–88, 2017.
- [YOI92] Yamato, J.; Ohya, J.; Ishii, K.: Recognizing human action in time-sequential images using hidden markov model. In: *Proceedings 1992 IEEE Computer Society conference on computer vision and pattern recognition*. S. 379–385, 1992.
- [Zä07] Zäh, M.; Wiesbeck, M.; Engstler, F.; Friesdorf, F.; Schubö, A.; Stork, S.; Banat, A.; Wallhoff, F.: Kognitive Assistenzsysteme in der manuellen Montage. In: *wt Werkstattstechnik online*. Bd. 97, S. 644–650, 2007.

- [Zh17] Zhang, W.; Ma, M.; Li, H.; Yu, J.: Generating interference matrices for automatic assembly sequence planning. *The International Journal of Advanced Manufacturing Technology* 90/1-4, S. 1187–1201, 2017.
- [ZLO07] Zimmermann, A.; Lorenz, A.; Oppermann, R.: An operational definition of context. In: *International and Interdisciplinary Conference on Modeling and Using Context*. S. 558–571, 2007.
- [Zu08] Zuehlke, D.: Smartfactory—from vision to reality in factory technologies. *IFAC Proceedings Volumes* 41/2, S. 14101–14108, 2008.
- [Zu10] Zuehlke, D.: SmartFactory—Towards a factory-of-things. *Annual reviews in control* 34/1, S. 129–138, 2010.
- [Zü16] Zühlke, D.: *The Rocky Road to the Factory of the Future*, 2016.
- [ZW08] Zäh, M. F.; Wiesbeck, M.: A model for adaptively generating assembly instructions using state-based graphs. In: *Manufacturing systems and technologies for the new frontier*. Springer, S. 195–198, 2008.

## Normen und Richtlinien

- [DIN 8580] DIN - Deutsches Institut für Normung e.V.: *Fertigungsverfahren - Begriffe, Einteilung*, Berlin, 2003.
- [ISO IEC 25010] International Organization for Standardization: *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*, Geneva, Switzerland, 2011.
- [VDI 2860] VDI-Gesellschaft Produktion und Logistik: *Montage- und Handhabungstechnik - Handhabungsfunktionen, Handhabungseinrichtungen, Begriffe, Definitionen, Symbole*, 1990.

## Hochschulschriften

- [Ae16] Aehnelt, M.: *Informationsassistent zur kognitiven Automatisierung manueller Montagearbeitsplätze*, Dissertation, Universität Rostock, 2016.
- [Ch04] Chen, H.: *An Intelligent Broker Architecture for Pervasive Context-Aware Systems*, Dissertation, University of Maryland, Baltimore County, 2004.
- [Fu16a] Funk, M.: *Augmented reality at the workplace: A context-aware assistive system using in-situ projection*, Dissertation, Stuttgart: Universität, 2016.

- [He03] Henricksen, K.: A framework for context-aware pervasive computing applications, Dissertation, Queensland: University of Queensland, 2003.
- [Ko14a] Korn, O.: Context-aware assistive systems for augmented work: A framework using gamification and projection, Dissertation, Stuttgart: Universität, 2014.
- [Lo13] Loskyll, M.: Entwicklung einer Methodik zur dynamischen kontextbasierten Orchestrierung semantischer Feldgerätefunktionalitäten, Dissertation, Technische Universität Kaiserslautern, 2013.
- [Rü14] Rütter, S.: Assistive systems for quality assurance by context-aware user interfaces in health care and production, Dissertation, Universitätsbibliothek Bielefeld, 2014.
- [We17a] Wendling, M.: Entwicklung und Realisierung einer Roboterzelle zur übergreifenden Interaktion mit benachbarten Fertigungsmodulen, Diplomarbeit, TU Kaiserslautern, 2017.
- [Wö14] Wölfle, M.: Kontextsensitive Arbeitsassistenzsysteme zur Informationsbereitstellung in der Intralogistik, Dissertation, Technische Universität München, 2014.

## Betreute studentische Arbeiten

- [Be19a] Bechtel, T.: Analyse und Bewertung von Assistenzsystemen hinsichtlich ihrer Nutzbarkeit für die manuelle Montage, Bachelorarbeit, Technische Universität Kaiserslautern, 2019.
- [Sc19] Schwing, M.: Entwicklung eines Verarbeitungsmodells für Assistenzsysteme zur kontextsensitiven Erkennung von Arbeitsschritten, Masterarbeit, Technische Universität Kaiserslautern, 2019.
- [Si17] Singh, A.: Modular and Adaptive Assistance System for Manual Assembly - Engineering a Semantically Described CPS Module, Masterarbeit, Technische Universität Kaiserslautern, 2017.
- [To19] Toshihiko, A.: Erkennung von Montagetätigkeiten mittels Smartwatch und künstlicher Intelligenz, Bachelorarbeit, Technische Universität Kaiserslautern, 2019.
- [Wa18] Wansch, A. H.: Konzept und Implementierung einer einheitlichen Sprache zur Beschreibung manueller Tätigkeiten, Bachelorarbeit, Technische Universität Kaiserslautern, 2018.

- [We20] Weber, M. B.: Evaluation der Eignung von Bewegungstrackern zur Erfassung von Arbeitsgesten als Grundlage für die Anwendung von Künstlicher Intelligenz, Forschungsprojekt, Technische Universität Kaiserslautern, 2020.

## Webseiten und Internetreferenzen

- [@DFKI 17] Deutsches Forschungszentrum für künstliche Intelligenz: DAKARA-Projekt: Design und Anwendung einer ultrakompakten, energieeffizienten und rekonfigurierbaren Kameramatrix zur Räumlichen Analyse, 2017, Stand: 06. 11. 2019.
- [@eCl@ss] eCl@ss: classification and product description, URL: <https://www.eclass.eu/>, Stand: 20. 10. 2019.
- [@ETIM] ETIM: ETIM Deutschland e.V. URL: [http : / / www . etim . de](http://www.etim.de), Stand: 20. 10. 2019.
- [@Fraunhofer] Jauch, C.: MonSiKo. Szenenanalyse in der assistierten Montage: Vortrag gehalten auf der Internationalen Fachmesse für Qualitätssicherung „Control“, 24.-27. April 2018, Stuttgart, hrsg. von Fraunhofer IPA, URL: <http://publica.fraunhofer.de/documents/N-523896.html>, Stand: 20. 11. 2019.
- [@MonSiKo] Projektblatt\_MonSiKo, URL: [https://www.softwaresysteme.pt-dlr.de/media/content/Projektblatt\\_MonSiKo.pdf](https://www.softwaresysteme.pt-dlr.de/media/content/Projektblatt_MonSiKo.pdf), Stand: 13. 09. 2019.
- [@motionEAP 1] motionEAP, URL: <http://www.motioneap.de/>, Stand: 13. 09. 2019.
- [@motionEAP 2] Projekt: motionEAP, URL: [https : / / www . digitale - technologien . de / DT / Redaktion / DE / Standardartikel / AutonomikFuerIndustrieProjekte / AutonomikFuerIndustrieProjekte / autonomik \\_ fuer \\_ industrie \\_ projekt - motioneap.html#content](https://www.digitale-technologien.de/DT/Redaktion/DE/Standardartikel/AutonomikFuerIndustrieProjekte/AutonomikFuerIndustrieProjekte/autonomik_fuer_industrie_projekt-motioneap.html#content), Stand: 26. 01. 2020.
- [@MQTT] MQTT - Message Queuing Telemetry Transport, URL: <http://mqtt.org/>, Stand: 08. 11. 2019.
- [@OMG] OMG: Business Process Model and Notation (BPMN), Version 2.0, URL: <https://www.omg.org/spec/BPMN/2.0/PDF>, Stand: 26. 10. 2019.
- [@OPTIMUM] OPTIMUM datamanagement solutions GmbH: Der schlaue Klaus - Montageunterstützung: Produktseite, URL: <https://www.optimum-gmbh.de/montage.html>, Stand: 18. 11. 2019.

- [@OWL] OWL Ostwestfalen Lippe GmbH: PROMIMO: Prozessintegrierte Mitarbeiter-Unterstützung in der Montage, 18.07.2018, URL: <https://www.owl-morgen.de/schaufenster/promimo/>, Stand: 03. 11. 2019.
- [@Rexroth 1] Bosch Rexroth AG: Assistenzsystem ActiveAssist: Produktkatalog, URL: [http://www.boschrexroth.com/various/utilities/mediadirectory/download/index.jsp?object\\_nr=R999001408](http://www.boschrexroth.com/various/utilities/mediadirectory/download/index.jsp?object_nr=R999001408), Stand: 25. 09. 2019.
- [@Rexroth 2] Bosch Rexroth AG: Assistenzsystem ActiveAssist: Produktseite, URL: <https://www.boschrexroth.com/de/de/produkte/produktgruppen/montagetechnik/news/assistenzsystem-activeassist/index>, Stand: 18. 11. 2019.
- [@Schnaithmann] Schnaithmann: 15\_Schnaithmann\_Flyer-cubuS\_DE, URL: [https://www.schnaithmann.de/fileadmin/user\\_upload/Mediathek/Innovationen/15\\_Schnaithmann\\_Flyer-cubuS\\_DE.pdf](https://www.schnaithmann.de/fileadmin/user_upload/Mediathek/Innovationen/15_Schnaithmann_Flyer-cubuS_DE.pdf), Stand: 13. 09. 2019.
- [@SF-KL 19] Technologie-Initiative SmartFactory KL: Neue Forschungsergebnisse im DAKARA-Projekt | SmartFactory-KL, 2019, URL: <https://smartfactory.de/neue-forschungsergebnisse-im-dakara-projekt/>, Stand: 06. 11. 2019.
- [@TNO] Bosch, T.; Van Thijn, G.: FAST, FLEXIBLE AND FAULTLESS ASSEMBLY WITH PROJECTED WORK INSTRUCTIONS: TNO operator support system, 2017, URL: <https://www.tno.nl/media/9326/tno-operator-support-systeem-assemblage17-8742.pdf>, Stand: 03. 11. 2019.
- [@Ulixes] Ulixes Robotersysteme GmbH: Der Assistent: Handbuch, URL: [http://derassistent.de/Content/Files/ulixes % 20 - %20Der % 20Assistent . %20Handbuch.pdf](http://derassistent.de/Content/Files/ulixes%20-%20Der%20Assistent.%20Handbuch.pdf), Stand: 18. 11. 2019.
- [@UNSPSC] UNSPSC: UNSPSC, URL: <https://www.unspsc.org/>, Stand: 20. 10. 2019.
- [@VDI] VDI: VDI 2860 - Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbol, URL: <https://www.vdi.de/richtlinien/details/vdi-2860-handhabungsfunktionen-handhabungseinrichtungen-begriffe-definitionen-symbol>, Stand: 20. 10. 2019.
- [Ja18] Jauch, C.: MonSiKo. Szenenanalyse in der assistierten Montage: Präsentation zum Vortrag auf der Internationalen Fachmesse für Qualitätssicherung „Control“, 24.-27. April 2018, Stuttgart, hrsg. von Fraunhofer IPA, 2018, URL: [http://publica.fraunhofer.de/eprints/urn\\_nbn\\_de\\_0011-n-5238969.pdf](http://publica.fraunhofer.de/eprints/urn_nbn_de_0011-n-5238969.pdf), Stand: 20. 11. 2019.
- [Wa17] Walz, J. D.: Ein adaptives Assistenzsystem für die Montage, hrsg. von Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, 2017, URL: <https://idw-online.de/de/news676725>, Stand: 26. 01. 2020.

# Lebenslauf

## Persönliche Daten

Name: Patrick Bertram  
Anschrift: Rosenstraße 35  
67655 Kaiserslautern  
Geburtsdatum: 25.09.1987  
Geburtsort: Rockenhausen  
Staatsangehörigkeit: deutsch

## Ausbildung

2009 - 2015 Technische Universität Kaiserslautern  
Studiengang: Elektrotechnik  
Vertiefung: Informationsverarbeitung  
2007 - 2009 Berufsbildende Schule I - Technik, Kaiserslautern  
Fachgebundene Hochschulreife

## Berufserfahrung

seit 2018 Deutsches Forschungszentrum für Künstliche Intelligenz GmbH -  
FB Innovative Fabrikssysteme /  
Technologie-Initiative SmartFactory KL e.V., Kaiserslautern  
*Researcher / Teamleiter Konstruktion und Entwicklung*  
2015 - 2018 Technologie-Initiative SmartFactory KL e.V., Kaiserslautern  
*Researcher*  
2012 - 2015 Fraunhofer, ITWM, Kaiserslautern  
Abteilung Bildverarbeitung  
*Hilfswissenschaftlicher Mitarbeiter*  
2004 - 2007 Meisterschule für Handwerker, Berufsfachschule, Kaiserslautern  
*Ausbildung zum Systemelektroniker*



# Werden Sie Autor im VDI Verlag!

## Publizieren Sie in „Fortschritt- Berichte VDI“



Veröffentlichen Sie die Ergebnisse Ihrer interdisziplinären technikorientierten Spitzenforschung in der renommierten Schriftenreihe **Fortschritt-Berichte VDI**. Ihre Dissertationen, Habilitationen und Forschungsberichte sind hier bestens platziert:

- **Kompetente Beratung und editorische Betreuung**
- **Vergabe einer ISBN-Nr.**
- **Verbreitung der Publikation im Buchhandel**
- **Wissenschaftliches Ansehen der Reihe Fortschritt-Berichte VDI**
- **Veröffentlichung mit Nähe zum VDI**
- **Zitierfähigkeit durch Aufnahme in einschlägige Bibliographien**
- **Präsenz in Fach-, Uni- und Landesbibliotheken**
- **Schnelle, einfache und kostengünstige Abwicklung**

**PROFITIEREN SIE VON UNSEREM RENOMMEE!**

[www.vdi-nachrichten.com/autorwerden](http://www.vdi-nachrichten.com/autorwerden)

vdI verlag

## Die Reihen der Fortschritt-Berichte VDI:

- 1 Konstruktionstechnik/Maschinenelemente
  - 2 Fertigungstechnik
  - 3 Verfahrenstechnik
  - 4 Bauingenieurwesen
- 5 Grund- und Werkstoffe/Kunststoffe
  - 6 Energietechnik
  - 7 Strömungstechnik
- 8 Mess-, Steuerungs- und Regelungstechnik
  - 9 Elektronik/Mikro- und Nanotechnik
  - 10 Informatik/Kommunikation
  - 11 Schwingungstechnik
- 12 Verkehrstechnik/Fahrzeugtechnik
  - 13 Fördertechnik/Logistik
- 14 Landtechnik/Lebensmitteltechnik
  - 15 Umwelttechnik
  - 16 Technik und Wirtschaft
  - 17 Biotechnik/Medizintechnik
  - 18 Mechanik/Bruchmechanik
  - 19 Wärmetechnik/Kältetechnik
- 20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
  - 21 Elektrotechnik
  - 22 Mensch-Maschine-Systeme
  - 23 Technische Gebäudeausrüstung

ISBN 978-3-18-304022-3