

**Forschungsbericht  
Institut für Automatisierungstechnik  
und Softwaresysteme**

Hrsg.: Prof. Dr.-Ing. Dr. h. c. Michael Weyrich

Timo Müller

**Selbstorganisiertes  
Rekonfigurationsmanagement  
cyber-physischer  
Produktionssysteme**

**Band 7/2025**

Universität Stuttgart

# **Selbstorganisiertes Rekonfigurationsmanagement cyber-physischer Produktionssysteme**

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik  
der Universität Stuttgart zur Erlangung der Würde eines  
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

Vorgelegt von  
Timo Müller  
aus Nagold

Hauptberichter: Prof. Dr.-Ing. Dr. h. c. Michael Weyrich

Mitberichter: Prof. Dr.-Ing. Thomas Bauernhansl

Tag der Einreichung: 14.06.2024

Tag der mündlichen Prüfung: 20.02.2025

Institut für Automatisierungstechnik und Softwaresysteme  
der Universität Stuttgart

2025



IAS-Forschungsberichte

Band 7/2025

**Timo Müller**

**Selbstorganisiertes Rekonfigurationsmanagement  
cyber-physischer Produktionssysteme**

D 93 (Diss. Universität Stuttgart)

Shaker Verlag  
Düren 2025

**Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Stuttgart, Univ., Diss., 2025

Copyright Shaker Verlag 2025

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

Print-ISBN        978-3-8440-9982-9  
PDF-ISBN        978-3-8191-0060-4  
ISSN              1610-4781  
eISSN            2944-8417  
<https://doi.org/10.2370/9783819100604>

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren  
Telefon: 02421 / 99 0 11 - 0 • Telefax: 02421 / 99 0 11 - 9  
Internet: [www.shaker.de](http://www.shaker.de) • E-Mail: [info@shaker.de](mailto:info@shaker.de)

Die vorliegende Arbeit entstand in weiten Teilen während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Automatisierungstechnik und Softwaresysteme (IAS) der Universität Stuttgart.

Mein besonderer Dank gilt meinem Doktorvater und Leiter des Instituts, Herrn Prof. Dr.-Ing. Dr. h.c. Michael Weyrich, für die Betreuung dieser wissenschaftlichen Arbeit, die Übernahme des Hauptberichts, sowie für die zahlreichen konstruktiven Diskussionen, wertvollen Hinweise und Anregungen.

Herrn Prof. Dr.-Ing. Thomas Bauernhansl danke ich für die Übernahme des Mitberichts und für das Interesse an meiner Arbeit.

Meinen Kolleginnen und Kollegen am IAS möchte ich für die Unterstützung, die familiäre Atmosphäre, die erstklassige Zusammenarbeit und die zahlreichen Diskussionen von Herzen danken. Ein ganz besonderer Dank gilt Herrn Dr.-Ing. Nasser Jazdi für die kontinuierliche Unterstützung bei der Erstellung dieser Arbeit, das wertvolle Feedback und die vielen (fachlichen) Gespräche. Für die kritische und gründliche Durchsicht des Manuskripts und die damit verbundenen Diskussionen danke ich Dominik Braun, Daniel Dittler und Simon Kamm.

Ebenso möchte ich mich bei den zahlreichen Studierenden bedanken, die im Zuge ihrer Abschlussarbeiten einen wertvollen Beitrag für das Gelingen dieser Arbeit geleistet haben.

Schließlich bin ich meiner Frau Kerstin und meinen beiden Töchtern Leonie und Emily dankbar, dass sie mich in dieser Zeit unterstützt haben, mir Motivation gaben und das notwendige geduldige Verständnis für mich aufbringen konnten.

Stuttgart, im März 2025

Timo Müller



## **Widmung**

Diese Arbeit widme ich meinen Töchtern Leonie und Emily und meiner Frau Kerstin.





# Inhaltsverzeichnis

|                                                                                                                  |           |
|------------------------------------------------------------------------------------------------------------------|-----------|
| Abbildungsverzeichnis .....                                                                                      | v         |
| Tabellenverzeichnis .....                                                                                        | viii      |
| Abkürzungsverzeichnis .....                                                                                      | ix        |
| Begriffsverzeichnis .....                                                                                        | xi        |
| Zusammenfassung .....                                                                                            | xiv       |
| Abstract .....                                                                                                   | xv        |
| <b>1 Einleitung .....</b>                                                                                        | <b>1</b>  |
| 1.1 Motivation und Problemstellung .....                                                                         | 1         |
| 1.2 Herausforderungen der Rekonfiguration im Stand der Technik .....                                             | 3         |
| 1.3 Ziele der Arbeit .....                                                                                       | 5         |
| 1.4 Wissenschaftliche Vorgehensweise .....                                                                       | 8         |
| 1.5 Aufbau der Arbeit .....                                                                                      | 10        |
| <b>2 Stand der Wissenschaft und Technik .....</b>                                                                | <b>13</b> |
| 2.1 Identifikation der Forschungslücke .....                                                                     | 13        |
| 2.2 Architekturen und Architekturmuster für cyber-physische Produktionssysteme .....                             | 17        |
| 2.3 Wissensmodellierung und Wissensverwaltung .....                                                              | 20        |
| 2.4 Rekonfigurationsmanagement .....                                                                             | 26        |
| 2.4.1 Ermittlung des Rekonfigurationsbedarfs .....                                                               | 26        |
| 2.4.2 Generierung alternativer Konfigurationen .....                                                             | 30        |
| 2.4.3 Bewertung und Auswahl von Konfigurationen .....                                                            | 39        |
| <b>3 Konzept für das selbstorganisierte Rekonfigurationsmanagement cyber-physischer Produktionssysteme .....</b> | <b>47</b> |
| 3.1 Übersicht über die Bestandteile des Konzepts .....                                                           | 47        |
| 3.2 Methodik für das selbstorganisierte Rekonfigurationsmanagement .....                                         | 48        |
| 3.2.1 Ermittlung des Rekonfigurationsbedarfs .....                                                               | 49        |
| 3.2.2 Generierung alternativer Konfigurationen .....                                                             | 51        |
| 3.2.3 Bewertung der Konfigurationen .....                                                                        | 60        |
| 3.2.4 Auswahl einer Konfiguration .....                                                                          | 60        |
| 3.3 Modellierung für das selbstorganisierte Rekonfigurationsmanagement .....                                     | 61        |
| 3.4 CPPS-Architektur für das selbstorganisierte Rekonfigurationsmanagement .....                                 | 65        |
| <b>4 Realisierung des Konzepts und des Tool-Supports .....</b>                                                   | <b>71</b> |
| 4.1 Realisierung des Management-Layers .....                                                                     | 71        |
| 4.2 Realisierung des Proxy-Layers .....                                                                          | 97        |
| 4.3 Realisierung des Control- und des Asset-Layers .....                                                         | 99        |
| 4.4 Systemimplementierung .....                                                                                  | 103       |

|                                             |                                                                    |            |
|---------------------------------------------|--------------------------------------------------------------------|------------|
| <b>5</b>                                    | <b>Evaluierung des Konzepts .....</b>                              | <b>104</b> |
| 5.1                                         | Ablauf der Evaluierung.....                                        | 104        |
| 5.2                                         | Definition von Evaluierungsszenarien.....                          | 105        |
| 5.3                                         | Durchführung der Evaluierungsszenarien .....                       | 107        |
| 5.4                                         | Auswertung der Ergebnisse der Evaluierungsszenarien.....           | 112        |
| 5.5                                         | Abgleich mit den definierten Anforderungen an die Konzeption ..... | 113        |
| <b>6</b>                                    | <b>Schlussbetrachtung .....</b>                                    | <b>117</b> |
| 6.1                                         | Zusammenfassung der Ergebnisse.....                                | 117        |
| 6.2                                         | Ausblick.....                                                      | 120        |
| <b>7</b>                                    | <b>Literaturverzeichnis.....</b>                                   | <b>123</b> |
| <b>Anhang A: Konzeptdetails .....</b>       |                                                                    | <b>132</b> |
| <b>Anhang B: Realisierungsdetails .....</b> |                                                                    | <b>136</b> |
| <b>Anhang C: Evaluierungsdetails.....</b>   |                                                                    | <b>147</b> |

# Abbildungsverzeichnis

|                                                                                                                            |    |
|----------------------------------------------------------------------------------------------------------------------------|----|
| Abbildung 1.1: Bestandteile des Rekonfigurationsmanagements.....                                                           | 2  |
| Abbildung 1.2: Angestrebte Resultate zur Beantwortung der drei Teilfragen .....                                            | 6  |
| Abbildung 1.3: Überblick: Design Science Research Methodik (vereinfacht und in Anlehnung an) [65, 70, 72].....             | 9  |
| Abbildung 1.4: Gliederung der Dissertationsschrift .....                                                                   | 12 |
| Abbildung 3.1: Bestandteile des Konzepts alias Teilartefakte .....                                                         | 48 |
| Abbildung 3.2: Rekonfigurationsmanagement-Methodik .....                                                                   | 49 |
| Abbildung 3.3: CPPS Capability Model (vereinfachtes Beispiel) [31] .....                                                   | 51 |
| Abbildung 3.4: Verfahren zur Generierung von Alternativen für Produktionssequenzen (Beispiel) [31] .....                   | 53 |
| Abbildung 3.5: Abbildung der Layout-Struktur des modularen Produktionssystems als Graph .....                              | 55 |
| Abbildung 3.6: Hierarchische Verknüpfung von Simulation und Optimierung nach Kategorie D - VDI 3633 (Blatt 12) [154] ..... | 56 |
| Abbildung 3.7: Modellierung des modularen Produktionssystems durch Zustandsautomaten .....                                 | 57 |
| Abbildung 3.8: Informationsmodell der CPMs [74] .....                                                                      | 62 |
| Abbildung 3.9: Informationsmodell der Produktionsaufträge [74].....                                                        | 64 |
| Abbildung 3.10: CPPS-Architektur für das selbstorganisierte Rekonfigurationsmanagement [74].....                           | 66 |
| Abbildung 4.1: Rekonfigurationsmanagement-GUI (RM-GUI).....                                                                | 72 |
| Abbildung 4.2: Programmablaufplan des Reconfiguration Demand Agent .....                                                   | 74 |
| Abbildung 4.3: Programmablaufplan eines System Configuration Agent.....                                                    | 76 |
| Abbildung 4.4: Darstellung eines Individuums (Layoutvariante).....                                                         | 77 |
| Abbildung 4.5: Vereinfachte Übersicht der Ermittlung von Layoutvarianten.....                                              | 79 |
| Abbildung 4.6: Programmablaufplan der CPPM Agenten .....                                                                   | 82 |
| Abbildung 4.7: Programmablaufplan des SBO Agent.....                                                                       | 83 |
| Abbildung 4.8: Kommunikation zwischen SBO Agent und MATLAB .....                                                           | 84 |
| Abbildung 4.9: Simulink Modell einer Systemkonfiguration des CPPS .....                                                    | 85 |
| Abbildung 4.10: Stateflow-Simulationsmodell einer Systemkonfiguration (Beispiel) [74].....                                 | 87 |
| Abbildung 4.11: Laufende Optimierung unter Einsatz eines Toolbox-basierten Simulated Annealing Algorithmus (Beispiel)..... | 88 |
| Abbildung 4.12: Laufende Optimierung unter Einsatz eines Toolbox-basierten Genetischen Algorithmus (Beispiel) .....        | 89 |
| Abbildung 4.13: Übersicht der Optimierung mit dem realisierten MOSA Algorithmus.....                                       | 90 |
| Abbildung 4.14: Auswahl der Kompromisslösung $\tilde{x}^*$ basierend auf der dritten Option .....                          | 95 |
| Abbildung 4.15: Programmablaufplan des Evaluation Agent .....                                                              | 97 |

|                                                                                                                                                |     |
|------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Abbildung 4.16: Workflow der EMF-basierten Modellierung und Erstellung von XML-Dateien [74].....                                               | 98  |
| Abbildung 4.17: Unity-Simulation des modularen Produktionssystems [74] .....                                                                   | 99  |
| Abbildung 4.18: Funktionsprinzip der eingesetzten SBC [74] .....                                                                               | 99  |
| Abbildung 4.19: Abarbeitung eines Produktionsauftrags durch die SBC (vereinfacht) .....                                                        | 101 |
| Abbildung 4.20: Webinterface der SBC - Produktionsauftragsübersicht .....                                                                      | 102 |
| Abbildung 5.1: Ablauf der Evaluierung .....                                                                                                    | 105 |
| Abbildung 5.2: Durchführung der Evaluierungsszenarien.....                                                                                     | 107 |
| Abbildung 5.3: Beispiele für unterschiedliche Startkonfigurationen des modularen Produktionssystems .....                                      | 107 |
| Abbildung 5.4: Übersicht von vier Produktionsaufträgen der Evaluierung.....                                                                    | 108 |
| Abbildung 5.5: Beispielhafte XML-Datei einer gefundenen alternativen Systemkonfiguration .....                                                 | 110 |
| Abbildung 5.6: Beispielhafte Transformation einer Startkonfiguration in eine neue Systemkonfiguration.....                                     | 111 |
| Abbildung 5.7: Beispiel für die resultierenden Gesamtaufwände nach Abschluss der Produktion in Unity .....                                     | 112 |
| Abbildung A.1: Universelle Methodik für die intelligente Exploration des Lösungsraums [153].....                                               | 132 |
| Abbildung A.2: Filterstufen innerhalb jedes Teil-Lösungsraums [153] .....                                                                      | 132 |
| Abbildung A.3: Intelligente Exploration des Lösungsraums für den Anwendungsfall des selbstorganisierten Rekonfigurationsmanagements [153]..... | 133 |
| Abbildung A.4: Detaillierter Überblick der Informationsmodellierung .....                                                                      | 134 |
| Abbildung A.5: Installations- und Deinstallationskosten eines Moduls im Produktionslayout .....                                                | 135 |
| Abbildung A.6: Installations- und Deinstallationszeit eines Moduls im Produktionslayout .....                                                  | 135 |
| Abbildung A.7: Installations- und Deinstallationsenergie eines Moduls im Produktionslayout .....                                               | 135 |
| Abbildung B.1: Subchart: Generator-Block.....                                                                                                  | 137 |
| Abbildung B.2: Subchart: CPTM-Block.....                                                                                                       | 137 |
| Abbildung B.3: Subchart: CPPM-Block .....                                                                                                      | 138 |
| Abbildung B.4: Subchart: Terminator-Block.....                                                                                                 | 138 |
| Abbildung B.5: Beispiel des initialisierten MATLAB Workspace einer Systemkonfiguration .....                                                   | 140 |
| Abbildung B.6: Zweidimensionale Visualisierung der Umgebung von $\mathbf{x} \in \Omega \subset \mathbb{R}^2$ .....                             | 140 |
| Abbildung B.7: Zweidimensionale Visualisierung des Einflusses von $\alpha$ auf die Akzeptanzwahrscheinlichkeit .....                           | 141 |
| Abbildung B.8: Zweidimensionale Visualisierung des Einflusses der Temperatur $c$ auf die Akzeptanzwahrscheinlichkeit .....                     | 141 |
| Abbildung B.9: Zweidimensionale Visualisierung des Einflusses der Gewichtung $\mathbf{w}_i$ auf die Akzeptanzwahrscheinlichkeit .....          | 142 |

|                                                                                                                                                                                                                                           |     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Abbildung B.10: Zweidimensionale Beispiele für die Auswahl von $\widetilde{\mathbf{x}}^*$ aus einer Pareto-Front für unterschiedliche Gewichtungen ( $\mathbf{w}_1, \mathbf{w}_2$ ) basierend auf der Methode der gewichteten Summe ..... | 142 |
| Abbildung B.11: Zweidimensionale Beispiele für die Auswahl von $\widetilde{\mathbf{x}}^*$ aus einer Pareto-Front für unterschiedliche Gewichtungen ( $\mathbf{w}_1, \mathbf{w}_2$ ) basierend auf dem Tchebycheff-Problem .....           | 143 |
| Abbildung B.12: Zwei Beispiele für eine konvexe Hülle um die Punktemenge einer Pareto-Front im zweidimensionalen Raum .....                                                                                                               | 143 |
| Abbildung B.13: Webinterface der SBC – Übersicht der Module im Steuerungsnetzwerk .....                                                                                                                                                   | 144 |
| Abbildung B.14: Webinterface der SBC – Detailansicht eines Produktionsmoduls im Steuerungsnetzwerk .....                                                                                                                                  | 144 |
| Abbildung B.15: Detaillierte Übersicht der Umsetzung des Control- und des Asset-Layers mit Anbindung des zugehörigen Webinterface zur Eingabe von Produktionsaufträgen und Überwachung des Produktionssystems .....                       | 145 |
| Abbildung B.16: Unity-GUI: Übersicht der Hauptansicht .....                                                                                                                                                                               | 145 |
| Abbildung B.17: Unity-GUI: Auswahl von Modulkonfigurationen welche per Drag & Drop ins Produktionslayout integriert werden können .....                                                                                                   | 146 |
| Abbildung B.18: Unity-GUI: Übersicht der Elemente für das Monitoring der Rekonfigurationsaufwände .....                                                                                                                                   | 146 |
| Abbildung C.1: Beschreibung des Produktionsauftrags <i>Distanzscheibe 1</i> und zugehörige XML-Datei .....                                                                                                                                | 149 |

## Tabellenverzeichnis

|              |                                                                                                                                              |     |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Tabelle 2.1: | Überblick der Ergebnisse der Literaturanalyse.....                                                                                           | 14  |
| Tabelle 2.2: | Interne und externe Trigger für die Rekonfiguration [130] .....                                                                              | 27  |
| Tabelle 5.1: | Überblick über die definierten Evaluierungsszenarien .....                                                                                   | 106 |
| Tabelle 5.2: | Ergebnisse der Evaluierungsszenarien.....                                                                                                    | 113 |
| Tabelle B.1: | Gewählte Parameter für die Realisierung der GA-basierten<br>Verfahrensvarianten in Abhängigkeit der Anzahl von eingesetzten CPPMs            | 136 |
| Tabelle B.2: | Rekonfigurationsmatrix: zeitliche Rekonfigurationsaufwände beim Überführen<br>in die verschiedenen Modulkonfigurationen des Bohrmoduls ..... | 136 |
| Tabelle C.1: | Installations- und Deinstallationsaufwände für die Komponenten der<br>Produktionsmodule .....                                                | 147 |

## Abkürzungsverzeichnis

|          |                                                                                                                                   |
|----------|-----------------------------------------------------------------------------------------------------------------------------------|
| BOP      | <b>B</b> ill of <b>P</b> rocess                                                                                                   |
| CPM      | Cyber- <b>p</b> hysisches <b>M</b> odul (engl.: Cyber- <b>p</b> hysical <b>m</b> odule)                                           |
| CPPM     | Cyber- <b>p</b> hysisches <b>P</b> roduktions <b>m</b> odul (engl.: Cyber- <b>p</b> hysical <b>p</b> roduction <b>m</b> odule)    |
| CPPS     | Cyber- <b>p</b> hysisches <b>P</b> roduktions <b>s</b> ystem (engl.: Cyber- <b>p</b> hysical <b>p</b> roduction <b>s</b> ystem)   |
| CPS      | Cyber- <b>p</b> hysisches <b>S</b> ystem (engl.: Cyber- <b>p</b> hysical <b>s</b> ystem)                                          |
| CPTM     | Cyber- <b>p</b> hysisches <b>T</b> ransport <b>m</b> odul (engl.: Cyber- <b>p</b> hysical <b>t</b> ransportation <b>m</b> odule)  |
| CSS      | Cascading Style Sheets                                                                                                            |
| DBMS     | <b>D</b> aten <b>b</b> ank <b>m</b> anagement <b>s</b> ystem                                                                      |
| DSR      | <b>D</b> esign <b>S</b> cience <b>R</b> esearch                                                                                   |
| DZ       | <b>D</b> igitaler <b>Z</b> willing                                                                                                |
| EMF      | <b>E</b> clipse <b>M</b> odeling <b>F</b> ramework                                                                                |
| FPB      | <b>F</b> ormalisierte <b>P</b> rozess <b>b</b> eschreibung                                                                        |
| FIPA     | <b>F</b> oundation for <b>I</b> ntelligent <b>P</b> hysical <b>A</b> gents                                                        |
| FIPA-ACL | <b>F</b> oundation for <b>I</b> ntelligent <b>P</b> hysical <b>A</b> gents – <b>A</b> gent <b>C</b> ommunication <b>L</b> anguage |
| GUI      | <b>G</b> raphical <b>U</b> ser <b>I</b> nterface                                                                                  |
| MAS      | <b>M</b> ultiagentensystem                                                                                                        |
| MAPE-K   | <b>M</b> onitor, <b>A</b> nalyze, <b>P</b> lan, <b>E</b> xecute und <b>K</b> nowledge                                             |
| MILP     | <b>M</b> ixed-integer <b>l</b> inear <b>p</b> rogramming                                                                          |
| MQTT     | <b>M</b> essage <b>Q</b> ueuing <b>T</b> elemetry <b>T</b> ransport                                                               |
| VDE      | <b>V</b> erband <b>d</b> er <b>E</b> lektrotechnik <b>E</b> lektronik <b>I</b> nformationstechnik                                 |
| VDI      | <b>V</b> erein <b>D</b> eutscher <b>I</b> ngenieure                                                                               |



|        |                                                                                                        |
|--------|--------------------------------------------------------------------------------------------------------|
| HTML   | <b>H</b> ypertext <b>M</b> arkup <b>L</b> anguage                                                      |
| HTTP   | <b>H</b> ypertext <b>T</b> ransfer <b>P</b> rotocol                                                    |
| IAS    | <b>I</b> nstitut für <b>A</b> utomatisierungstechnik und <b>S</b> oftwaresysteme                       |
| I4.0   | <b>I</b> ndustrie <b>4.0</b>                                                                           |
| JADE   | <b>J</b> ava <b>A</b> gent <b>D</b> Evelopment <b>F</b> ramework                                       |
| ODP    | <b>O</b> ntology <b>D</b> esign <b>P</b> attern                                                        |
| OPC    | <b>O</b> pen <b>P</b> latform <b>C</b> ommunications                                                   |
| OPC UA | <b>O</b> PC <b>U</b> nified <b>A</b> rchitecture                                                       |
| OWL    | <b>W</b> eb <b>O</b> ntology <b>L</b> anguage                                                          |
| PPR    | <b>P</b> rodukt, <b>P</b> rozess, <b>R</b> essource                                                    |
| RDF    | <b>R</b> esource <b>D</b> escription <b>F</b> ramework                                                 |
| RMS    | <b>R</b> econfigurable <b>M</b> anufacturing <b>S</b> ystems                                           |
| RMT    | <b>R</b> econfigurable <b>M</b> achine <b>T</b> ool                                                    |
| SBC    | <b>S</b> ervice-oriented <b>A</b> rchitecture <b>b</b> ased <b>C</b> ontrol                            |
| SOA    | <b>S</b> ervice-orientierte <b>A</b> rchitektur (engl.: <b>S</b> ervice-oriented <b>A</b> rchitecture) |
| SPARQL | <b>S</b> PARQL <b>P</b> rotocol <b>A</b> nd <b>R</b> DF <b>Q</b> uery <b>L</b> anguag                  |
| SWRL   | <b>S</b> emantic <b>W</b> eb <b>R</b> ule <b>L</b> anguage                                             |
| TAD    | <b>T</b> ool <b>A</b> pproach <b>D</b> irection                                                        |
| TCP/IP | <b>T</b> ransmission <b>C</b> ontrol <b>P</b> rotocol/ <b>I</b> nternet <b>P</b> rotocol               |
| TOS    | <b>T</b> ool <b>O</b> rientation <b>S</b> pace                                                         |
| UML    | <b>U</b> nified <b>M</b> odeling <b>L</b> anguage                                                      |
| VWS    | <b>V</b> erwaltungsschale                                                                              |
| XML    | <b>E</b> xtensible <b>M</b> arkup <b>L</b> anguage                                                     |

## Begriffsverzeichnis

**Agent:** Ein Agent ist eine abgrenzbare Einheit, die durch eigenständiges Verhalten definierte Ziele erreicht. Dabei interagiert er mit anderen Agenten und seiner Umgebung. [1]

**Architektur:** „Kombination von Elementen eines Modells aufbauend auf Prinzipien und Regeln zum Zweck seiner Konstruktion, Weiterentwicklung und Nutzung.“ [2]

**Asset:** Gegenstand, der einen Wert für eine Organisation hat. [2]

**Bill of Process:** Beinhaltet die Sequenz der auszuführenden Schritte, die zur Fertigung eines Produkts benötigt werden.

**Cyber-physische Produktionssysteme:** Produktionssysteme, die aus cyber-physischen Systemen bestehen, werden auch als cyber-physische Produktionssysteme bezeichnet. [3]

**Cyber-physische Systeme:** „Sind gekennzeichnet durch eine Verknüpfung von realen (physischen) Objekten und Prozessen mit informationsverarbeitenden (virtuellen) Objekten und Prozessen über offene, teilweise globale und jederzeit miteinander verbundene Informationsnetze.“ [4]

**FIPA-ACL:** Die Agent Communication Language dient zur Kommunikation zwischen Agenten eines Multiagentensystems und ist eine Spezifikation der Foundation for Intelligent Physical Agents zur Gewährleistung der Interoperabilität durch die Bereitstellung eines Standardsatzes von ACL-Nachrichtenstrukturen und eines wohldefinierten Verfahrens zur Pflege dieses Satzes. [5]

**Genetische Algorithmen:** Suchalgorithmen, die sich an den Grundprinzipien der biologischen Evolution und der natürlichen Selektion orientieren. Sie simulieren die Evolution lebender Organismen, bei der die stärkeren Individuen die schwächeren dominieren, indem sie die biologischen Mechanismen der Evolution wie Selektion, Kreuzung und Mutation nachahmen. [6]

**Konfiguration:** Eine Konfiguration bildet den aktuellen Zustand eines Systems ab.

**Middleware:** Eine Softwareschicht oberhalb des Betriebssystems, aber unterhalb des Anwendungsprogramms, die eine gemeinsame Programmierabstraktion in einem verteilten System bietet. [7]

**Modularität:** Die Modularität eines Systems gibt an, inwieweit ein System in jeweils in sich geschlossene Bausteine zerlegt und gekapselt ist. [8]

**Modell:** „Vereinfachte Nachbildung eines geplanten oder existierenden Systems mit seinen Prozessen in einem anderen begrifflichen oder gegenständlichen System.“ [9]

**Multiagentensystem:** System aus mehreren Agenten, die gemeinsam ein übergeordnetes Problem lösen.

**Referenzarchitektur:** „Modell für eine Architekturbeschreibung (für I4.0), die allgemein genutzt wird und als zweckmäßig anerkannt ist (Referenzcharakter hat).“ [2]

**Rekonfiguration:** „Eine Rekonfiguration stellt die technische Sicht des Prozesses der Veränderung eines bereits entwickelten und operativ eingesetzten Systems dar, um es an neue Anforderungen anzupassen, Funktionalität zu erweitern, Fehler zu beseitigen oder die Qualitätseigenschaften zu verbessern.“ [10]

**Rekonfigurierbare Produktionssysteme:** Produktionssysteme, die von vornherein für schnelle Änderungen ihrer Struktur sowie ihrer Hardware- und Softwarekomponenten ausgelegt sind, um die Produktionskapazität und Funktionalität schnell an sich ändernde Anforderungen anpassen zu können. [11]

**Rekonfigurierbarkeit:** „Unter Rekonfigurierbarkeit wird im Allgemeinen die Möglichkeit zur Anpassung einer Produktionsressource an veränderte Bedingungen mittels Austauschen, Hinzufügen oder Eliminieren von Komponenten oder Baugruppen verstanden.“ [12]

**Reconfigurable Machine Tools (RMTs):** Sind nach [13] eine zentrale Komponente rekonfigurierbarer Produktionssysteme und können in verschiedene Konfigurationen überführt werden, um anpassbare Funktionalität und Kapazität zu bieten, indem ihre Basismodule beibehalten werden, während das Hinzufügen, Entfernen oder Modifizieren von Hilfsmodulen möglich ist.

**Service: (im Deutschen: Dienst)** „Abgegrenzter Funktionsumfang, der von einer Entität oder Organisation über Schnittstellen angeboten wird.“ [2]

**Service-orientierte Architektur:** Paradigma für die Organisation und Nutzung verteilter Funktionalität. [14]

**Simulation:** „Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind. Insbesondere werden die Prozesse über die Zeit entwickelt.“ [9]

**Simulationsmodell:** „Stellen vereinfachte Abbilder einer Realität dar und verhalten sich bezüglich der untersuchungsrelevanten Aspekte weitgehend analog zum geplanten oder realen System. In Erweiterung dieses Begriffes handelt es sich bei Simulationsmodellen

um ablauffähige Modelle zur experimentellen Analyse der dynamischen Zusammenhänge in Systemen.“ [15]

**Simulated Annealing:** Simulated Annealing ist eine probabilistische Methode, um das globale Minimum einer Kostenfunktion zu finden, die mehrere lokale Minima aufweisen kann. Sie funktioniert durch Nachahmung des physikalischen Prozesses, bei dem ein Festkörper langsam abgekühlt wird, so dass seine Struktur schließlich bei einer Konfiguration mit minimaler Energie "eingefroren" wird. [16]

**System:** „Eine von ihrer Umwelt abgegrenzte Menge von Elementen, die miteinander in Beziehung stehen.“ [9]

**Verwaltungsschale:** „Virtuelle digitale und aktive Repräsentanz einer I4.0-Komponente im I4.0-System.“ [2]

## Zusammenfassung

Aufgrund kürzerer Innovations- und Produktlebenszyklen sowie wirtschaftlicher Volatilität wird die Rekonfiguration von Produktionssystemen während der Betriebsphase immer häufiger erforderlich. Ein weiterer Trend in der industriellen Automatisierung ist das Aufkommen von cyber-physischen Produktionssystemen, die vielversprechende Potenziale wie z.B. die Fähigkeit zur Selbstorganisation bieten. In den seltenen Fällen, in denen heute bereits eine Rekonfiguration von Produktionssystemen realisiert wird, geschieht dies in einem fehleranfälligen und zeitaufwändigen Prozess. Dieser wird überwiegend manuell und individuell durchgeführt und ist in hohem Maße von der Erfahrung des Menschen abhängig. Dies wird jedoch dem oft enorm großen Lösungsraum nicht gerecht und führt dazu, dass die Entscheidung, wie eine Rekonfiguration durchgeführt werden soll, nicht nach objektiven Kriterien erfolgt und zudem nur eine Teilmenge möglicher Lösungen betrachtet wird.

Infolgedessen wird in dieser Arbeit die Zielsetzung abgeleitet und verfolgt, cyber-physische Produktionssysteme um die Fähigkeit eines selbstorganisierten Rekonfigurationsmanagements zu bereichern. Basierend auf hergeleiteten Anforderungen an ein geeignetes Konzept wird die Forschungslücke im Stand der Wissenschaft und Technik aufgezeigt, um den Bedarf dieser Forschungsarbeit zu untermauern. Das vorgestellte Konzept beinhaltet eine Methodik für das selbstorganisierte Rekonfigurationsmanagement, einen Ansatz zur Wissensmodellierung und -verwaltung sowie eine Architektur für cyber-physische Produktionssysteme. Die Methodik umfasst dazu die Ausprägung der vier Schritte *Ermittlung des Rekonfigurationsbedarfs*, *Generierung alternativer Konfigurationen*, *Bewertung der Konfigurationen* und *Auswahl einer Konfiguration* und basiert auf einer Modellierung die insbesondere auf der Formalisierten Prozessbeschreibung beruht und mittels UML/XML umgesetzt wird. Insgesamt wird ein dezentrales Verfahren eingesetzt, um den Lösungsraum alternativer Systemkonfigurationen systematisch aufzuspannen und simulationsbasiert multikriteriell zu optimieren. Hierzu wurden verschiedene Varianten des Simulated Annealing und des Genetischen Algorithmus umgesetzt. Die Architektur integriert den Ansatz zur Wissensmodellierung und -verwaltung sowie die Rekonfigurationsmanagement-Methodik und nutzt ein Multiagentensystem sowie eine service-orientierte Architektur.

Basierend auf der prototypischen Realisierung eines modularen Produktionssystems mit Matrix-Layout wurde das Konzept erfolgreich evaluiert. Dabei wurde ein breites Spektrum an Evaluierungsszenarien untersucht, um die Funktionalität für diverse Produktionsaufträge sowie Komponentenausfälle, variierende Komplexität und unterschiedliche Start-Konfigurationen aufzuzeigen.

Zudem kam das Design Science Research als übergeordnete Forschungsmethodik zum Einsatz.

**Schlagwörter:** Rekonfigurationsmanagement, Cyber-physische Produktionssysteme, Industrielle Automatisierung, Wissensmodellierung

## Abstract

Due to shorter innovation and product life cycles as well as economic volatility, reconfiguration of production systems during the operational phase is becoming increasingly necessary. Another development in industrial automation is the emergence of cyber-physical production systems, offering promising potentials such as self-organization capabilities. Nowadays, the reconfiguration of production systems is rarely realized and is done in an error-prone and time-consuming process. This process is predominantly carried out manually and individually and is highly dependent on human experience. As a result, the decision as to how a reconfiguration should be carried out is not based on objective criteria and, moreover, only a subset of possible solutions is considered.

Consequently, this thesis derives and pursues the objective of enhancing cyber-physical production systems with a self-organized reconfiguration management. Based on derived requirements for a suitable concept, the research gap in the state of science and technology is identified to substantiate the necessity of this research work. Subsequently, the concept is presented, which includes a methodology for self-organized reconfiguration management, an approach for knowledge modeling and management, and an architecture for cyber-physical production systems. The methodology comprises the specification of the four steps *identification of reconfiguration demand*, *generation of alternative configurations*, *evaluation of configurations* and *selection of a new configuration* and is based on a modeling which is particularly derived from the Formalized Process Description and implemented using UML/XML. Overall, a decentralized procedure is used to systematically explore the solution space of alternative system configurations and to optimize them in a simulation-based, multi-objective manner. For this purpose, different variants of the simulated annealing and the genetic algorithm were implemented. The architecture integrates the approach for knowledge modeling and management as well as the reconfiguration management methodology and uses a multi-agent system and a service-oriented architecture.

Based on the prototypical realization of a modular production system with matrix layout, the concept was successfully evaluated. A wide range of evaluation scenarios was investigated to demonstrate the functionality for diverse production orders as well as component failures, varying complexity and different start configurations.

In addition, Design Science Research was used as an overarching research methodology.

**Keywords:** Reconfiguration Management, Cyber-physical Production Systems, Industrial Automation, Knowledge Modeling



# 1 Einleitung

„Nichts ist so beständig wie der Wandel“ - dieses Zitat von Heraklit lässt sich auch auf die gegenwärtige und zukünftige Situation in der industriellen Automatisierung übertragen und kann als abstrakte Motivation dieser Forschungsarbeit interpretiert werden.

## 1.1 Motivation und Problemstellung

Bedingt durch zunehmend kürze Innovations- und Produktlebenszyklen sowie der wirtschaftlichen Volatilität treten Anforderungsänderungen an Produktionssysteme zur Betriebszeit immer häufiger auf [17–27]. Es wird demzufolge immer schwieriger, gar unmöglich, eine Vorhersage aller Ziele eines Produktionssystems bereits zur Entwicklungszeit leisten zu können [28]. Daher werden Produktionssysteme, unabhängig von der Größe ihres (zur Entwurfszeit ausgelegten) Flexibilitätsskorridors, die benötigte Funktionalität schlussendlich nicht bereitstellen können. Es ergibt sich Rekonfigurationsbedarf zur Betriebszeit, welcher die zentrale Problemstellung, die in dieser Forschungsarbeit adressiert wird, darstellt.

Die Definition „Eine Rekonfiguration stellt die technische Sicht des Prozesses der Veränderung eines bereits entwickelten und operativ eingesetzten Systems dar, um es an neue Anforderungen anzupassen, Funktionalität zu erweitern, Fehler zu beseitigen oder die Qualitätseigenschaften zu verbessern.“ [10], wird in dieser Arbeit zugrunde gelegt und lässt sich derart zusammenfassen, dass unter einer Rekonfiguration jegliche Anpassung eines bereits operativ eingesetzten Systems verstanden wird.

Um zu verdeutlichen, dass das Thema Rekonfiguration mehr als nur **die Durchführung von Rekonfigurationsmaßnahmen** umfasst, wurde der Begriff des **Rekonfigurationsmanagements** (siehe Abbildung 1.1) eingeführt. Hierzu wurde aufbauend auf [29] in [30–32] erläutert, dass das Rekonfigurationsmanagement die Schritte der **Ermittlung des Rekonfigurationsbedarfs** sowie der **Rekonfigurationsplanung** umfasst<sup>1</sup>. Dabei kann die Rekonfigurationsplanung weiter in die Schritte der **Generierung alternativer Konfigurationen**, der **Bewertung der Konfigurationen** sowie der **Auswahl einer Konfiguration** unterteilt werden. Die Durchführung von Rekonfigurationsmaßnahmen ist transparent abgebildet, da sie als optionale Erweiterung angesehen wird, die klassischerweise manuell durchgeführt wird und nicht im Fokus dieser Forschungsarbeit steht.

---

<sup>1</sup> Die Publikationen [30–32] sind im Rahmen dieser Forschungsarbeit entstanden.





**Abbildung 1.1: Bestandteile des Rekonfigurationsmanagements**

Zudem wird die Zukunft der industriellen Automatisierung vom Konzept der cyber-physischen Systeme (CPS) geprägt sein [3, 33–40], das eine Schlüsselrolle in Industrie 4.0-Architekturen einnimmt [41, 42]. Kernaspekte solcher CPS sind (basierend auf [4]), neben ihren physischen Komponenten, ihre Konnektivität und ihre Fähigkeit zur Informationsverarbeitung. Diese ermöglichen es CPS, einen Grad an Intelligenz innezuhaben, der in seinen konkreten Ausprägungen stark variieren kann. Produktionssysteme, die aus CPS bestehen, werden hierbei auch als cyber-physische Produktionssysteme (CPPS) bezeichnet [3, 43]. Hierzu ist anzumerken, dass die Produktionssysteme der Zukunft nach [23] aus cyber-physischen Produktionsmodulen (CPPMs) bestehen, die dem Produktionsnetzwerk ihre Fähigkeiten als Services zur Verfügung stellen.

Durch ihre hohe Vernetzung stellen CPPS die „Vision von adaptiven, sich selbstkonfigurierenden und teilweise selbstorganisierenden, flexiblen Produktionsanlagen“ [39, 44] dar und können zu geringeren Rüstzeiten und einem optimiertem Energie- und Ressourceneinsatz [39, 44] führen. Um jedoch die Potenziale von CPPS, wie beispielsweise ein selbstorganisiertes Rekonfigurationsmanagement erschließen zu können, müssen zunächst Voraussetzungen wie die Existenz von semantischen Beschreibungen des CPPS und seiner Komponenten erfüllt sein. In diesem Zusammenhang kam [45] zu dem Schluss, dass die Ausschöpfung des vollen Potenzials von CPS semantische Modelle erfordert. Aufbauend auf diesen semantischen Informationen, können entsprechende Konzepte entwickelt und in die Anwendung überführt werden. Darüber hinaus besteht ein Bedarf an Konzepten, die bestehende Algorithmen aus Domänen wie der Künstlichen Intelligenz oder der mathematischen Optimierung, gegebenenfalls in angepasster Form, übernehmen [44].

## 1.2 Herausforderungen der Rekonfiguration im Stand der Technik

Die CPPS haben mehrere Wurzeln innerhalb der Produktionsdomäne, einige davon sind Intelligente Produktionssysteme (IMS), rekonfigurierbare Produktionssysteme (RMS), biologische Produktionssysteme (BMS) und digitale Fabriken (DF) [46]. Folglich soll das Konzept der CPPS nicht im Widerspruch zu diesen Entwicklungen stehen, sondern diese, zumindest teilweise, zusammenführen und weiter verbessern [47]. Da der steigende Rekonfigurationsbedarf zur Betriebszeit als zentrale Problemstellung dieser Forschungsarbeit identifiziert wurde, ist hierbei insbesondere das Konzept der RMS [11] relevant.

Basierend auf der Literatur über RMS wie beispielsweise [48–50] weisen diese fünf wesentliche Rekonfigurations-Aspekte auf, welche rekonfigurierbar sein sollen und somit auch für die Rekonfiguration berücksichtigt werden sollen. Diese Rekonfigurations-Aspekte sind nachfolgend aufgelistet und kurz beschrieben [48–50]:

- **System:** Dieser Aspekt betrifft die Auswahl und Anordnung der Module, d.h. der Maschinen, innerhalb des Layouts.
- **Software:** Die Softwarearchitektur sollte modular, modifizierbar, erweiterbar und wiederverwendbar sein und verschiedene Konfigurationen umfassen können.
- **Steuerung:** Die Steuerungsstruktur sollte nicht angepasst werden müssen. Dies bedeutet, dass das Steuerungskonzept auch nach jeder Rekonfiguration noch greifen soll.
- **Prozess:** Die Anpassung des Produktionsprozesses im Hinblick auf eine optimale Performance, um die geforderte Menge in der geforderten Qualität produzieren zu können, sollte möglich sein (z.B. Einstellung der Prozessparameter).
- **Maschine:** Die Hardwarestruktur einer Maschine sollte rekonfigurierbar sein, z.B. um alternative Funktionen anzubieten oder die Produktionsrate anzupassen.

Der als Problemstellung spezifizierte steigende Rekonfigurationsbedarf wird, neben vielen anderen, auch in [51–55] genannt. Obwohl die wirtschaftliche und ökologische Notwendigkeit der Rekonfiguration von Produktionssystemen unumstritten ist, wird sie bisher kaum realisiert [52, 56]. Dies ist darauf zurückzuführen, dass die Rekonfiguration von Produktionssystemen derzeit folgende Defizite aufweist [17, 23, 57–59]:

- Sie ist **zeitaufwändig**, da sie manuell und individuell ausgelöst, geplant und durchgeführt wird.
- Sie ist **fehleranfällig**, weil sie nicht systematisch oder methodisch unterstützt wird.
- Es gibt keine Garantie für eine optimumnahe Lösung, da sich auf begrenztes menschliches Wissen gestützt wird und daher **nur eine Teilmenge des Lösungsraums für alternative Konfigurationen berücksichtigt wird**.
- Die **Bewertung und Auswahl** finden **nicht anhand von (mehreren) objektiven Kriterien** statt, sondern basieren auf Erfahrungswerten.

Diese Defizite sollen mit Hilfe eines Rekonfigurationsmanagements adressiert werden.

Des Weiteren hängen der Erfolg und die Akzeptanz von RMS nach [56] und [58] in erster Linie vom Aufwand für die Rekonfiguration des Systems und dem daraus resultierenden Nutzen ab. Um den Aufwand zu reduzieren, kann zum einen eine erhöhte Rekonfigurierbarkeit des Systems angestrebt werden. Zum anderen kann versucht werden, die notwendigen manuellen Tätigkeiten des Rekonfigurationsmanagements zu reduzieren. Darüber hinaus können sowohl die Minimierung der Rekonfigurationsaufwände, welche zur Überführung der aktuellen Konfiguration eines Produktionssystems in eine alternative Konfiguration anfallen, als auch der Produktionsaufwände, welche bei der Durchführung eines Produktionsauftrags anfallen, angestrebt werden.

Zur Erhöhung der Rekonfigurierbarkeit lassen sich nach [12] und [60] im Stand der Forschung die folgenden sechs Eigenschaften [61] identifizieren, welche RMS sowie deren Bestandteile innehaben sollten: Modularität, Integrationsfähigkeit, Umrüstbarkeit, Anpassungsfähigkeit, Diagnosefähigkeit und Skalierbarkeit.

Die Erhöhung der Rekonfigurierbarkeit, teilweise bis hin zur sogenannten Plug & Produce-Fähigkeit, ist Gegenstand vieler vergangener und aktueller Forschungsbemühungen [22, 24, 38, 62, 63], steht jedoch nicht im Fokus dieser Forschungsarbeit. Das jeweilige Maß an Rekonfigurierbarkeit stellt vielmehr eine Gegebenheit dar, welche berücksichtigt werden soll.

## 1.3 Ziele der Arbeit

Ziel dieser Forschungsarbeit ist es demnach ein Rekonfigurationsmanagement zu konzipieren welches die oben genannten Aspekte berücksichtigt und die genannten Defizite adressiert. Um dieses Ziel zu erreichen, müssen zunächst aber auch Kenntnisse über die aktuelle Konfiguration des Produktionssystems als notwendige Voraussetzung vorhanden sein.

In diesem Zusammenhang wird in dieser Forschungsarbeit die These untersucht, dass CPPS vielversprechende Potenziale bieten, um ein Rekonfigurationsmanagement durchführen zu können. Die Formulierung dieser These begründet sich insbesondere darauf, dass CPPS über Modelle verfügen, Intelligenz bieten können und (intern) vernetzt sind. So können sowohl die Automatisierung des Rekonfigurationsmanagements als auch die Identifizierung der aktuellen Konfiguration des Produktionssystems durch die Potenziale, die CPPS bieten, angegangen werden.

Daraus ergibt sich die Zielsetzung dieser Forschungsarbeit als:

*Cyber-physische Produktionssysteme sollen um die Fähigkeit zu einem selbstorganisierten Rekonfigurationsmanagement bereichert werden.*

Umformuliert als Forschungsfrage lautet diese:

*Wie können cyber-physische Produktionssysteme um die Fähigkeit eines selbstorganisierten Rekonfigurationsmanagements bereichert werden?*

Um diese Forschungsfrage beantworten zu können, gilt es die nachfolgenden drei Teilfragen zu beantworten, welche eingeführt werden, um die Beantwortung der formulierten Forschungsfrage wie auch die dafür notwendige Konzeption zu strukturieren:

- 1) Mit welcher Methodik kann ein selbstorganisiertes Rekonfigurationsmanagement ermöglicht werden?
- 2) Wie können Ressourcen und Dienste hierfür geeignet beschrieben werden?
- 3) Wie können die Bestandteile dieser Methodik im CPPS verortet und koordiniert werden?

Die erste Teilfrage beschäftigt sich offensichtlich mit dem Entwurf einer geeigneten **Methodik** und stellt die wichtigste Teilfrage dar. Die zweite Teilfrage hingegen ist der Findung einer entsprechenden **Modellierung** gewidmet. Zuletzt adressiert die dritte Teilfrage die Verwendung einer zweckmäßigen **CPPS-Architektur**, die sowohl die Wissensmodellierung und –verwaltung, als auch die Methodik integriert. Diese angestrebten Resultate zur Beantwortung der drei Teilfragen sind in Abbildung 1.2 visualisiert.

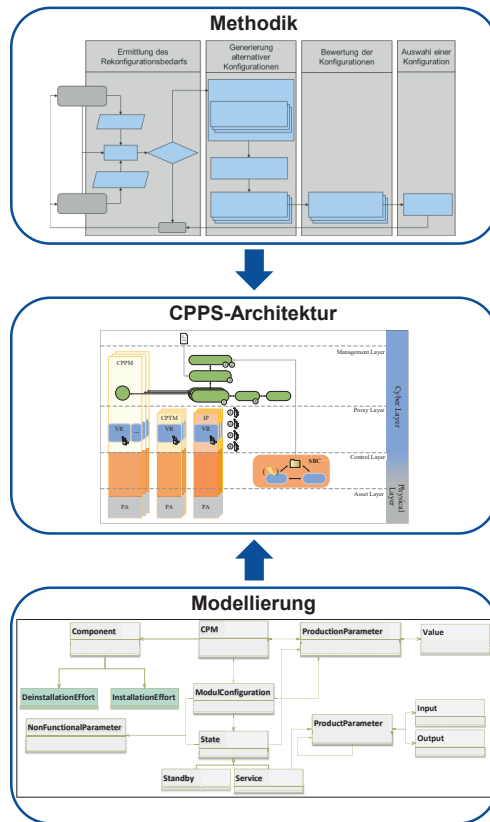


Abbildung 1.2: Angestrebte Resultate zur Beantwortung der drei Teilfragen

Die aus Kapitel 1.1 und 1.2 hergeleiteten **Anforderungen an die Konzeption** sind im nachfolgenden angeführt und kurz beschrieben:

**Anforderung 1 (A1):** Geeignete Abbildung der Schritte des Rekonfigurationsmanagements im CPPS

Entsprechend der in Kapitel 1.1 eingeführten Bestandteile des Rekonfigurationsmanagements, wird eine geeignete Abbildung der Schritte **Ermittlung des Rekonfigurationsbedarfs (A1.1)**, **Generierung alternativer Konfigurationen (A1.2)**, **Bewertung der Konfigurationen (A1.3)** sowie **Auswahl einer Konfiguration (A1.4)** benötigt. Daraus ergibt sich nachfolgend deren Abbildung als eigenständige Teilanforderungen um die jeweils relevanten Aspekte und Defizite aus Kapitel 1.2 verorten zu können. Somit soll die in Kapitel 1.2 angeführte Fehleranfälligkeit durch eine systematische und methodische Vorgehensweise adressiert werden.

### **Anforderung 1.1 (A1.1):** Ermittlung des Rekonfigurationsbedarfs

Für die Ermittlung des Rekonfigurationsbedarfs muss ein Abgleich der jeweils aktuellen Produktionsanforderungen mit der aktuellen Konfiguration des Produktionssystems durchgeführt werden. Hierzu müssen auch geeignete Trigger identifiziert werden, welche den genannten Abgleich auslösen.

### **Anforderung 1.2 (A1.2):** Generierung alternativer Konfigurationen

Die Generierung alternativer Konfigurationen dient der Erzeugung von Konfigurationsalternativen des Produktionssystems, welche die jeweils aktuellen Produktionsanforderungen umsetzen können. Diese soll *auf der Maschinenebene die Hardware und Software umfassen* (A1.2.1) und *auf der Systemebene die Auswahl von Maschinen, deren Positionierung, sowie die Anpassung des Produktionsprozesses (also dessen Sequenz, Allokation und Parametrierung) umfassen* (A1.2.2). Somit werden über A1.2.1 und A1.2.2 die zugehörigen, in Kapitel 1.2 genannten, Rekonfigurations-Aspekte adressiert. Um zu verhindern das nur eine (willkürliche) Teilmenge des Lösungsraums für alternative Konfigurationen berücksichtigt wird, soll eine *Intelligente Exploration des Lösungsraums von Konfigurationsalternativen realisiert werden* (A1.2.3).

### **Anforderung 1.3 (A1.3):** Bewertung der Konfigurationen

Die Bewertung der entsprechend A1.2 generierten alternativen Konfigurationen soll sowohl *den Rekonfigurationsaufwand berücksichtigen* (A1.3.1) als auch *den Produktionsaufwand berücksichtigen* (A1.3.2). Dies geschieht um dem in Kapitel 1.2 als wesentlich angeführten Verhältnis aus Aufwand und Nutzen Rechnung zu tragen. Des Weiteren soll die Bewertung der alternativen Konfigurationen *multikriteriell und anhand objektiver Kriterien erfolgen* (A1.3.3) um zum einen zu verhindern das diese lediglich auf Erfahrungswerten basiert und zum anderen die Inklusion verschiedener, als wichtig erachteter Kriterien, zu ermöglichen.

### **Anforderung 1.4 (A1.4):** Auswahl einer Konfiguration

Um die Schritte des Rekonfigurationsmanagements abzurunden, soll zuletzt auch eine Auswahl einer geeigneten Konfigurationsalternative des Produktionssystems basierend auf A1.3 erfolgen. Hierzu muss eine zweckmäßige Beschreibung der ausgewählten Konfiguration erfolgen, sodass deren anschließende Anwendung ermöglicht wird.

### **Anforderung 2 (A2):** Automatisierte Durchführung des Rekonfigurationsmanagements

Um sowohl die Fehleranfälligkeit als auch den implizierten zeitlichen Aufwand, welche in den Defiziten (s. Kapitel 1.2) angeführt wurden, zu minimieren, soll eine automatisierte Durchführung des Rekonfigurationsmanagements realisiert werden.

### Anforderung 3 (A3): Nutzung der Potenziale von cyber-physischen Produktionssystemen

Basierend auf der Beschreibung der CPPS und des Konzepts der CPS im Allgemeinen (s. Kapitel 1.1) soll die zu Beginn dieses Kapitels aufgestellte These, dass CPPS vielversprechende Potenziale bieten, um ein Rekonfigurationsmanagement durchführen zu können, untersucht werden. Daher sollen die Potenziale von CPPS genutzt werden um die These anschließend belegen zu können, bzw. eine argumentative Diskussion zu dieser zu ermöglichen.

## 1.4 Wissenschaftliche Vorgehensweise

Um ein angestrebtes Forschungsvorhaben methodisch bestmöglich zu unterstützen, muss ein Ansatz gewählt werden, der zur Zielsetzung und dem jeweiligen Forschungsbereich passt.

Basierend auf [64] kann eine Vorgehensweise, die darauf abzielt eine (konkrete) Lösung anzubieten, dazu beitragen, die bestehende Kluft zwischen Theorie und Praxis zu verringern. Darüber hinaus wird die Anwendung der Forschung vereinfacht und die Anerkennung ihrer Relevanz für die Praxis gefördert [64].

Diesbezüglich positioniert sich das *Design Science Research (DSR)* als ein epistemologisches Paradigma, das die Forschung in Richtung Problemlösung [65] und Artefaktdesign lenkt [66]. Das DSR dient also dazu, die Grenzen traditioneller Ansätze beim Entwerfen, Konstruieren oder Schaffen eines neuen Artefakts, d. h. von etwas, das noch nicht existiert, oder bei der Durchführung von Forschungsarbeiten mit Schwerpunkt auf die Problemlösung zu überwinden [66]. Außerdem will das DSR nicht nur Wissen anwenden, sondern auch Wissen schaffen [66]. Jedoch zielt das DSR (wie z.B. auch das Operations Research), obwohl es sich auf die Problemlösung fokussiert, nicht auf optimale sondern zufriedenstellende Ergebnisse ab [66]. Die generierten Lösungen sollen jedoch für eine bestimmte Klasse von Problemen verallgemeinbar sein [64, 67–69].

Nach [70] ist das DSR, ein **systematischer Prozess der Gestaltung von Artefakten zur Problemlösung**, um zu evaluieren was entworfen wurde oder was funktioniert und der Kommunikation der Ergebnisse.

Abbildung 1.3 bietet einen Überblick über das DSR nach Hevner [71]. Das iterative Vorgehen wird dabei durch die drei Zyklen Relevanz, Design und Rigor in Verbindung mit den drei Säulen Umwelt, DSR und Wissensbasis umgesetzt. Die drei Zyklen werden nachfolgend kurz und mit Bezug auf die vorliegende Forschungsarbeit erläutert, wobei zur Vermeidung von Redundanz eine Zuordnung zwischen DSR und den Inhalten der Dissertationsschrift in Kapitel 1.5 erfolgt und weitere Details in den jeweiligen Kapiteln aufgeführt werden: [71–73]

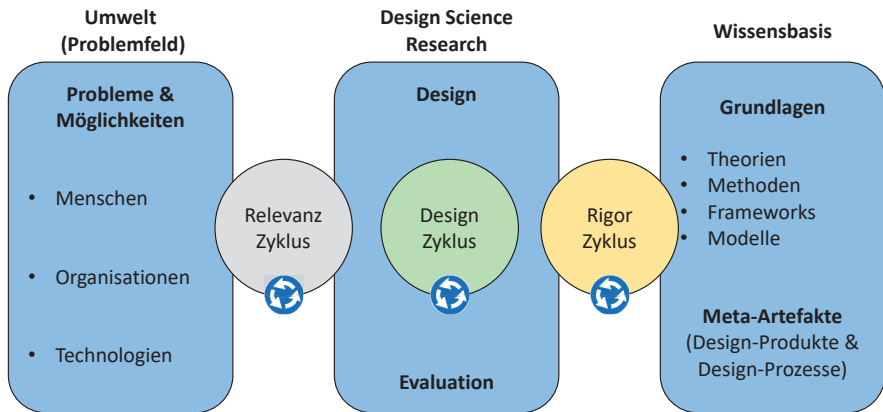


Abbildung 1.3: Überblick: Design Science Research Methodik (vereinfacht und in Anlehnung an) [66, 71, 73]

**Relevanz Zyklus:** Im Rahmen des *Relevanz Zyklus* wird zunächst, basierend auf der Betrachtung des Problemfelds, die Problemstellung identifiziert, die Forschungsfrage formuliert und die Anforderungen an das zu entwerfende Artefakt abgeleitet. Abschließend werden die Ergebnisse der Evaluation (Design Zyklus) dazu genutzt um zu beurteilen ob die formulierte Forschungsfrage hinreichend beantwortet wurde.

Im Rahmen dieser Dissertation erfolgte diesbezüglich nach der initialen Identifikation der Problemstellung und der Herausforderungen sowie der initialen Formulierung der Zielsetzung und der der Forschungsfrage und der Ableitung der Anforderungen an das zu entwerfende Artefakt deren sukzessive Weiterentwicklung. Diese Weiterentwicklung erfolgte iterativ und insbesondere durch die Betrachtung des Problemfelds der Rekonfiguration von cyber-physischen Produktionssystemen und den Austausch mit Fachexperten. Somit ergab sich eine stetige Weiterentwicklung der Anforderungen als Ausgangsbasis für die einzelnen Iterationen des Design Zyklus. Schlussendlich resultierten die Tätigkeiten des Relevanz Zyklus in den oben angeführten Herausforderungen und der korrespondierenden Zielsetzung mit der zugehörigen Forschungsfrage und den Anforderungen an die Konzeption. Die damit verbundene Einführung der drei Teilfragen der formulierten Forschungsfrage bildet die Grundlage für die Entwicklung der drei Teilartefakte (Methodik, Modellierung und CPPS-Architektur) im Rahmen des Design Zyklus.

**Design Zyklus:** Der *Design Zyklus* bildet den Kern des Design Science Researchs und beschäftigt sich mit der Artefakterstellung durch einen iterativen Prozess des Entwurfs (design) des Artefakts. Der Design Zyklus umfasst dabei neben der Identifikation des zu entwerfenden Artefakts, die Entwicklung von Entwurfsalternativen und die Evaluierung der Alternativen anhand der Anforderungen, bis ein zufriedenstellender Entwurf erreicht ist. Der Design Zyklus integriert dabei sowohl Erkenntnisse aus dem Relevanz Zyklus als auch aus dem Rigor Zyklus.



Im Kontext dieser Dissertation wurden mehrere Iterationen des Design Zyklus durchgeführt, um die drei Teilartefakte durch die Tätigkeiten der Konzepterstellung und Realisierung auszuprägen und anschließend gegenüber den Anforderungen zu evaluieren. Die Iterationen wurden dabei insbesondere durch Erkenntnisse aus dem Rigor Zyklus im Sinne von Literaturrecherchen, aber auch durch die beschriebenen Weiterentwicklungen der Anforderungen getrieben. Weitere Treiber waren die Fertigstellung einzelner studentischer Arbeiten und deren kritische Analyse sowie der Austausch mit Fachexperten.

**Rigor Zyklus:** Der *Rigor Zyklus* bindet die Wissensbasis, im Sinne des Stands der Forschung und Technik, in das DSR ein. Hierdurch können bereits bestehende wissenschaftliche Erkenntnisse, etwa im Sinne von Theorien und Methoden oder auch im Sinne sogenannter Meta-Artefakte (Design-Produkte und -Prozesse) in den Design Zyklus integriert werden. Darüber hinaus findet über den Rigor Zyklus zum einen die Abgrenzung zur existierenden Wissensbasis statt um den Innovationsgehalt der Arbeit sicherzustellen. Zum anderen sollen der Wissensbasis neue wissenschaftliche Erkenntnisse, insbesondere auch in verallgemeinerter Form, hinzugefügt werden (z.B. in Form von Publikationen mit wissenschaftlicher Qualitätssicherung).

Im Hinblick auf den Rigor Zyklus fanden im Rahmen dieser Dissertation insbesondere mehrere Literaturrecherchen statt. So wurde zur Identifikation der Forschungslücke zunächst eine qualitative Betrachtung des internationalen Stands der Wissenschaft und Technik durchgeführt. Anschließend erfolgte eine strukturierte Literaturrecherche mit Fokus auf den Kreis der TuLAUT (Theorie und Lehre der Automatisierungstechnik<sup>2</sup>). Um den Forschungsbedarf anhand der abgeleiteten Anforderungen nochmals final abzusichern und weiteren Erkenntnisgewinn zu generieren, wurde zuletzt eine umfangreiche systematische Literaturanalyse im Rahmen einer internationalen Kooperation durchgeführt (hierzu sei auf Kapitel 2 verwiesen). Darüber hinaus wurden im Zuge dieser Dissertation zahlreiche Publikationen veröffentlicht, um die Wissensbasis fortwährend mit neuen wissenschaftlichen Erkenntnissen anzureichern.

## 1.5 Aufbau der Arbeit

Die vorliegende Forschungsarbeit gliedert sich in 6 Kapitel (s. Abbildung 1.4). Während die durchgeführten Forschungsaktivitäten entsprechend des Design Science Researchs in mehreren Iterationen stattfanden, wird hier lediglich die thematische Zuordnung der angeführten Inhalte der Dissertationsschrift selbst zu den drei Zyklen des Design Science Researchs beschrieben.

---

<sup>2</sup> <http://tulaut.net/>

Kapitel 1 hat bereits die Problemstellung spezifiziert, die Herausforderungen im Stand der Technik aufgeführt und nach der Formulierung der Forschungsfrage die zugehörigen Anforderungen abgeleitet (Relevanz Zyklus). Darüber hinaus werden die zu entwerfenden Artefakte und Teilartefakte definiert (Design Zyklus).

Kapitel 2 umfasst die Abbildung des relevanten Stands der Wissenschaft und Technik. Hierdurch können sowohl Ansätze für den späteren Entwurf der definierten (Teil-)Artefakte als auch die bestehende Forschungslücke identifiziert werden (Rigor Zyklus).

Kapitel 3 beschreibt die Konzeption für das selbstorganisierte Rekonfigurationsmanagement cyber-physischer Produktionssysteme und stellt das wesentliche zu entwerfende Artefakt dieses Forschungsvorhabens dar. Hier wird zunächst eine Übersicht über die Bestandteile des Konzepts, welche den in Abbildung 1.2 dargestellten angestrebten Resultaten zur Beantwortung der drei Teilfragen der Forschungsfrage entsprechen, gegeben. Diese bilden im Sinne des Design Science Researchs drei zu entwickelnde Teilartefakte und werden anschließend detailliert beschrieben (Design Zyklus).

Kapitel 4 umfasst mit der Realisierung des Konzepts das zweite Artefakt dieser Arbeit und ist anhand der Realisierung der einzelnen Layer der CPPS-Architektur strukturiert. Darüber hinaus wird der Tool-Support für die Modellerstellung beschrieben, welcher mit Hilfe des Eclipse Modeling Frameworks realisiert wurde (Design Zyklus).

Das Kapitel 5 widmet sich der Evaluierung des Konzepts. Hierzu wird zunächst der Ablauf der Evaluierung vorgestellt. Basierend darauf folgen die jeweiligen Teilkapitel zur Definition, Durchführung und Auswertung der Ergebnisse der Evaluierungsszenarien. Abschließend erfolgt der Abgleich mit den definierten Anforderungen an die Konzeption. Somit werden in Kapitel 5 zunächst dem Design Zyklus zugehörige Tätigkeiten durchgeführt (Evaluation), zuletzt erfolgt aber auch eine Beurteilung, ob die formulierte Forschungsfrage hinreichend beantwortet wurde, welche dem Relevanz Zyklus zugehörig ist.

Abschließend erfolgt in Kapitel 6 eine Schlussbetrachtung, die sowohl die Ergebnisse der Forschungsarbeit zusammenfasst als auch einen Ausblick auf weitere mögliche (wissenschaftliche) Aktivitäten gibt. Dieses Kapitel ist dem Rigor Zyklus zugeordnet. An dieser Stelle angemerkt, dass sowohl die (komplette) Dissertationsschrift selbst, als auch die Publikationen die zu den einzelnen Inhalten der Dissertationsschrift veröffentlicht wurden, eine Erweiterung der Wissensbasis darstellen und somit dem Rigor Zyklus zugeordnet werden können.

| Gliederung                                                                                                          |  |
|---------------------------------------------------------------------------------------------------------------------|--|
| <b>Kapitel 1: Einleitung</b>                                                                                        |  |
| 1.1 Motivation und Problemstellung                                                                                  |  |
| 1.2 Herausforderungen der Rekonfiguration im Stand der Technik                                                      |  |
| 1.3 Ziele der Arbeit                                                                                                |  |
| 1.4 Wissenschaftliche Vorgehensweise                                                                                |  |
| 1.5 Aufbau der Arbeit                                                                                               |  |
| <b>Kapitel 2: Stand der Wissenschaft und Technik</b>                                                                |  |
| 2.1 Identifikation der Forschungslücke                                                                              |  |
| 2.2 Architekturen und Architekturmuster für cyber-physische Produktionssysteme                                      |  |
| 2.3 Wissensmodellierung und Wissensverwaltung                                                                       |  |
| 2.4 Rekonfigurationsmanagement                                                                                      |  |
| <b>Kapitel 3: Konzept für das selbstorganisierte Rekonfigurationsmanagement cyber-physischer Produktionssysteme</b> |  |
| 3.1 Übersicht über die Bestandteile des Konzepts                                                                    |  |
| 3.2 Methodik für das selbstorganisierte Rekonfigurationsmanagement                                                  |  |
| 3.3 Modellierung für das selbstorganisierte Rekonfigurationsmanagement                                              |  |
| 3.4 CPPS-Architektur für das selbstorganisierte Rekonfigurationsmanagement                                          |  |
| <b>Kapitel 4: Realisierung des Konzepts und des Tool-Supports</b>                                                   |  |
| 4.1 Realisierung des Management-Layers                                                                              |  |
| 4.2 Realisierung des Proxy-Layers                                                                                   |  |
| 4.3 Realisierung des Control- und des Asset-Layers                                                                  |  |
| 4.4 Systemimplementierung                                                                                           |  |
| <b>Kapitel 5: Evaluierung des Konzepts</b>                                                                          |  |
| 5.1 Ablauf der Evaluierung                                                                                          |  |
| 5.2 Definition von Evaluierungsszenarien                                                                            |  |
| 5.3 Durchführung der Evaluierungsszenarien                                                                          |  |
| 5.4 Auswertung der Ergebnisse der Evaluierungsszenarien                                                             |  |
| 5.5 Abgleich mit den definierten Anforderungen an die Konzeption                                                    |  |
| <b>Kapitel 6: Schlussbetrachtung</b>                                                                                |  |
| 6.1 Zusammenfassung der Ergebnisse                                                                                  |  |
| 6.2 Ausblick                                                                                                        |  |

Abbildung 1.4: Gliederung der Dissertationsschrift

## 2 Stand der Wissenschaft und Technik

Wie in Kapitel 1 erläutert wurde, ist die Betrachtung des Stands der Wissenschaft und Technik dem Rigor Zyklus des DSR zugeordnet. In Kapitel 1 wurde im Rahmen des Relevanz Zyklus des DSR über die Betrachtung des Problemfeldes die Problemstellung identifiziert, die Forschungsfrage formuliert und die Anforderungen an das zu entwerfende Artefakt abgeleitet. Über diese Anforderungen ist es nun möglich, die Forschungslücke im Stand der Wissenschaft und Technik zu identifizieren, um so den bestehenden Forschungsbedarf abzusichern. In Kapitel 2.1 erfolgt daher zunächst die genannte Identifikation der Forschungslücke, bevor in Kapitel 2.2 ein Überblick über Architekturen und Architekturmuster für CPPS gegeben wird. Anschließend wird in Kapitel 2.3 auf die Wissensmodellierung und -verwaltung im Kontext des Rekonfigurationsmanagements eingegangen. Abschließend werden in Kapitel 2.4 bestehende Ansätze im Hinblick auf methodische Umsetzung der Bestandteile des Rekonfigurationsmanagements (vgl. Kapitel 1.1) erläutert. Einige Inhalte dieses Kapitels wurden zudem in bereits veröffentlichten Publikationen thematisiert, insbesondere sei hier auf [74] in Bezug auf Kapitel 2.1 und Kapitel 2.4, sowie auf [75] in Bezug auf Kapitel 2.2 und Kapitel 2.3 verwiesen<sup>3</sup>.

### 2.1 Identifikation der Forschungslücke

Zur Identifikation der Forschungslücke, die durch diese Forschungsarbeit geschlossen werden soll, wird nachfolgend das Ergebnis einer quantitativen Literaturanalyse präsentiert. Hierzu wurden die angeführten Publikationen hinsichtlich der Erfüllung der in Kapitel 1.3 abgeleiteten Anforderungen A1-A3 und deren Unteranforderungen bewertet. Der Überblick über eine Auswahl der analysierten Publikationen ist in Tabelle 2.1 zusammengefasst, die möglichen Einstufungen des Erfüllungsgrades der Anforderungen kann dabei der Legende der Tabelle entnommen werden. Für A2 wurde dabei eine Einstufung anhand des Level of Automation (LoA) vorgenommen, wie unten weiter ausgeführt wird. Im Rahmen einer internationalen Kooperation wurden in einer systematischen Literaturanalyse 404 Publikationen gesichtet, von denen 67 Publikationen detailliert untersucht und hinsichtlich der Erfüllung der Anforderungen dieser Forschungsarbeit sowie weiterführender Aspekte bewertet wurden. Die nachfolgend vorgestellten Ergebnisse werden somit durch [74] untermauert, wobei dort auch zusätzliche Aspekte und gewonnene Erkenntnisse beschrieben werden.

---

<sup>3</sup> Die genannten Publikationen sind im Rahmen dieser Forschungsarbeit entstanden.

Tabelle 2.1: Überblick der Ergebnisse der Literaturanalyse

| #    | A1 | A1.1 | A1.2 | A1.2.1 | A1.2.2 | A1.2.3 | A1.3 | A1.3.1 | A1.3.2 | A1.3.3 | A1.4 | A2 | A3 |
|------|----|------|------|--------|--------|--------|------|--------|--------|--------|------|----|----|
| [76] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [77] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [78] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [79] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [80] |    |      |      |        |        |        |      |        |        |        |      | 3  |    |
| [81] |    |      |      |        |        |        |      |        |        |        |      | 3  |    |
| [82] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [83] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [84] |    |      |      |        |        |        |      |        |        |        |      | 3  |    |
| [85] |    |      |      |        |        |        |      |        |        |        |      | 3  |    |
| [86] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [87] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [88] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [89] |    |      |      |        |        |        |      |        |        |        |      | 10 |    |
| [90] |    |      |      |        |        |        |      |        |        |        |      | 3  |    |
| [23] |    |      |      |        |        |        |      |        |        |        |      | 1  |    |
| [91] |    |      |      |        |        |        |      |        |        |        |      | 1  |    |

Legende:

- (annähernd) nicht erfüllt
- zu einem kleinen Teil erfüllt
- teilweise erfüllt
- zu einem großen Teil erfüllt
- (annähernd) vollständig erfüllt

In Bezug auf **Anforderung 1 (A1)** „Geeignete Abbildung der Schritte des Rekonfigurationsmanagements im CPPS“ lässt sich festhalten, dass keiner der untersuchten Ansätze alle Schritte in vollem Umfang adressiert. Es ist außerdem ersichtlich, dass viele Ansätze sich auf jeweils ausgewählte Schritte fokussieren.

Die Beurteilung von **Anforderung 1.1 (A1.1)** „Ermittlung des Rekonfigurationsbedarfs“ verdeutlicht, dass dieser Schritt in den meisten Ansätzen vernachlässigt wird. Dies ist in der Regel darauf zurückzuführen, dass häufig die Neuplanung eines Systems adressiert wird oder die Existenz eines Rekonfigurationsbedarfs als gegeben angenommen wird. Sofern eine Ermittlung des Rekonfigurationsbedarfs stattfindet, werden häufig entweder externe Trigger [84, 87], wie etwa ein neuer Produktionsauftrag, oder interne Trigger [78, 88], wie ein Komponentenausfall, berücksichtigt. Nur in wenigen der untersuchten Ansätze, wie etwa in [76], wird beiden Trigger-Arten Rechnung getragen.

Auch in Bezug auf **Anforderung 1.2 (A1.2)** „Generierung alternativer Konfigurationen“ zeigt die Literaturrecherche im Allgemeinen auf, dass sich viele Ansätze lediglich auf die Durchführung einer Produktionsplanung und –steuerung von Grund auf, anstatt auf die Betrachtung eines bestehenden Produktionssystems und dessen Rekonfiguration konzentrieren. Zu den inkludierten Unteranforderungen von A1.2 lässt sich zunächst festhalten, dass **Anforderung 1.2.1 (A1.2.1)**, das Rekonfigurationsmanagement soll „auf der Maschinenebene die Hardware und Software umfassen“, von vielen der untersuchten Ansätze zumindest teilweise erfüllt wird. Oft geschieht dies jedoch nicht in vollem Umfang, so weist keiner der untersuchten Ansätze eine Inklusion von alternativen Hardware- und Software-Alternativen auf. Bei den Ansätzen, die lediglich einen der beiden Aspekte umfassen, liegt der Schwerpunkt zumeist auf dem Hardware-Aspekt oder der Software-Aspekt ist zumindest nicht klar adressiert worden (beispielsweise [79–82]). Nur wenige Ansätze (wie etwa [76] oder [86]) fokussieren sich auf den Software-Aspekt. **Anforderung 1.2.2 (A1.2.2)**, das Rekonfigurationsmanagement soll „auf der Systemebene die Auswahl von Maschinen, deren Positionierung, sowie die Anpassung des Produktionsprozesses (also dessen Sequenz, Allokation und Parametrierung) umfassen“, wird von vielen der analysierten Ansätze teilweise erfüllt. Insgesamt lässt sich hierbei festhalten, dass viele Ansätze die Auswahl von Maschinen sowie die Anpassung des Produktionsprozesses umfassen, wohingegen die Positionierung von ausgewählten Maschinen im Layout des Produktionssystems fast immer vernachlässigt wird. Obwohl die Anpassung des Produktionsprozesses oft adressiert wird, ist anzumerken, dass keiner der untersuchten Ansätze alle drei genannten Aspekte Sequenz, Allokation und Parametrierung umfasst. Von diesen Ansätzen integrieren die meisten lediglich eine Allokation der Prozessschritte zu den Produktionsressourcen [82, 87, 90]. In einigen Fällen wird zudem eine Anpassung bzw. die Bildung von Alternativen der Produktionssequenz adressiert [23, 88, 89], der Aspekt der Parametrierung wird jedoch nur sehr selten (z.B. in [86]) aufgegriffen. Hinsichtlich **Anforderung 1.2.3 (A1.2.3)** „Intelligente Exploration des Lösungsraums von Konfigurationsalternativen“

wurde untersucht, ob die Ansätze den Lösungsraum über eine Zerlegung des Problems in mehrere, kleinere Teilprobleme sowie durch eine intelligente Reduktion des Lösungsraumes explorieren. Diese beiden Aspekte wurden nur sehr selten innerhalb eines einzelnen Ansatzes realisiert. Für [86] wurde A1.2.3 als erfüllt eingestuft, obwohl im Detail eine sehr spezifische Umsetzung beschrieben wird. So zielt [86] nicht auf eine globale Optimierung ab, sondern eher auf ein globales Management auf Systemebene. Dennoch reduzieren die Autoren den Lösungsraum, indem sie nicht gangbare, nicht sinnvolle sowie ungünstige Lösungen verwerfen, um sicherzustellen, dass das System in der Lage ist, den Produktionsauftrag auszuführen. Darüber hinaus wird auch der Aspekt der Aufteilung erfüllt, da die Exploration des gesamten Lösungsraum aufgeteilt wird, indem Ressourcenagenten für die Generierung ihrer eigenen optimalen Rekonfigurationsalternativen verantwortlich sind. Die Literaturanalyse hat gezeigt, dass es sowohl einige Ansätze gibt, die diese Anforderung überhaupt nicht adressieren [85, 90, 91], als auch einige die lediglich einen der beiden Aspekte einbeziehen [83, 87, 89]. Unter den Ansätzen, die lediglich einen Aspekt berücksichtigen, liegt der Fokus fast ausschließlich auf dem Aspekt der Reduktion des Lösungsraums, welcher dann mit Hilfe von Constraints für ein mathematisches Modell verwirklicht wird.

Die in **Anforderung 1.3 (A1.3)** geforderte „Bewertung der Konfigurationen“ wird in den begutachteten Ansätzen hinsichtlich des Einbezugs der drei Teilanforderungen, dass das Rekonfigurationsmanagement „den Rekonfigurationsaufwand berücksichtigen“ (A1.3.1), „den Produktionsaufwand berücksichtigen“ (A1.3.2) und „multikriteriell und anhand objektiver Kriterien erfolgen“ (A1.3.3) soll, sehr unterschiedlich eingebunden. So wird A1.3 beispielsweise von [76, 77] und [91] überhaupt nicht erfüllt. Einige der Ansätze erfüllen hingegen jeweils eine der Untieranforderungen. So werden etwa in [78] und [84] der Rekonfigurationsaufwand berücksichtigt (A1.3.1), wohingegen in [83] und [88] der Produktionsaufwand (A1.3.2) berücksichtigt wird. Eine etwas größere Anzahl der Ansätze erfüllen sogar zugleich zwei der Teilanforderungen, wobei hier primär die Kombination aus A1.3.1 und A1.3.2 [80, 82, 87, 89] auftritt und etwas seltener die Kombination aus A1.3.1 und A1.3.3 [79, 85]. Am wenigsten häufig werden alle drei Teilanforderungen A1.3.1, A1.3.2 und A1.3.3 zugleich erfüllt, dies ist in [81, 86] der Fall. Insgesamt lässt sich im Rahmen der analysierten Literatur hierzu festhalten, dass von den drei Teilanforderungen A1.3.1 (der Rekonfigurationsaufwand) am häufigsten adressiert wird. Etwas weniger häufig wird A1.3.2 (der Produktionsaufwand) adressiert und A1.3.3 (die multikriterielle Bewertung) wird am seltensten integriert.

Bei der Bewertung des Erfüllungsgrades der **Anforderung 1.4 (A1.4)** „Auswahl einer Konfiguration“ werden neben dem Fall, dass diese überhaupt nicht erfolgt, zwei Ausprägungen unterschieden. In der ersten Ausprägung wird dem Betreiber des Produktionssystems eine Auswahl alternativer Konfigurationen angeboten, aus denen er selektiert. In der zweiten Ausprägung wird hingegen eine alternative Konfiguration ausgewählt, die für die im jeweiligen Ansatz formulierten Ziele als am besten geeignet angesehen wird. Daher besteht bezüglich dieser Anforderung und

ihrer Bewertung eine starke Kopplung an den Umfang des jeweiligen Ansatzes. Infolgedessen weist eine ganze Reihe der untersuchten Ansätze die zweite Ausprägung auf. Deutlich weniger oft ist die erste Ausprägung vertreten, beispielsweise in [80, 81, 84, 90]. Ebenso haben nur wenige Ansätze wie [23, 85] und [91] den Aspekt der Auswahl einer Konfiguration komplett vernachlässigt.

Die **Anforderung 2 (A2)** „Automatisierte Durchführung des Rekonfigurationsmanagements“ wurde basierend auf dem Level of Automation (LoA) nach [92] beurteilt. Das LoA wird in aufsteigender Reihenfolge vergeben, beginnend mit LoA=1, d.h. der Computer bietet keinerlei Unterstützung, bis hin zu LoA=10, d.h. der Computer entscheidet alles selbstständig. Es ist zu beachten, dass der Schritt des Rekonfigurationsmanagements mit dem niedrigsten zugewiesenen LoA-Wert den gesamten LoA-Wert definiert. Für den Fall, dass nicht alle Schritte im jeweiligen Ansatz behandelt wurden, werden nur die berücksichtigten Schritte in Betracht gezogen. Daher wurde der größte Anteil der Ansätze (z.B. [76, 79, 86, 87]) mit LoA=10 bewertet. In einigen Fällen wie [80, 87] und [85] wurde ein LoA=3 festgestellt, wohingegen ein LoA=1 nur selten [23, 91] auftritt.

Bezüglich **Anforderung 3 (A3)** „Nutzung der Potenziale von cyber-physischen Produktionssystemen“ wurde festgestellt, dass die meisten der analysierten Ansätze [80, 82, 83, 85] diese Potenziale überhaupt nicht nutzen. Einige Ansätze schöpfen entweder durch die Verwendung von Modellen [23, 78] oder durch die Nutzung des Vernetzungsaspekts [87, 88] die Potenziale von CPPS teilweise aus. Darüber hinaus werden bei einigen Ansätzen beide Potenziale zumindest in gewisser Weise genutzt [76, 77, 91].

*Insgesamt ergab die Untersuchung des Stands der Wissenschaft und Technik, dass die formulierte Forschungsfrage nicht hinreichend beantwortet wird. Somit wurde die Forschungslücke identifiziert und der Bedarf an dieser Forschungsarbeit untermauert.*

## 2.2 Architekturen und Architekturmuster für cyber-physische Produktionssysteme

Da es sich bei der Emergenz von CPPS um eine evolutionäre Entwicklung handelt, gibt es einige etablierte Grundprinzipien, die auch bei der Realisierung dieser Systeme und ihrer Architektur berücksichtigt werden sollten. Daher sind Prinzipien wie die *(De-)Komposition* (Divide and Conquer), die die *Trennung von Verantwortlichkeiten* (Separation of Concerns) beinhaltet, von fundamentaler Bedeutung [8]. Zusammen mit dem Prinzip des *Verbergen von Informationen* (Information Hiding), das die Verwendung klar definierter Schnittstellen vorsieht, wird damit das Prinzip der Modularität umgesetzt [8]. Das Prinzip der Modularität ist insbesondere, aber nicht



ausschließlich, im Zusammenhang mit der Rekonfiguration wichtig, da es die Rekonfigurierbarkeit eines Systems erhöht. Darüber hinaus sollten auch einige Techniken für ein gutes Design in Betracht gezogen werden, die als SOLID-Prinzipien, d. h. das Prinzip der eindeutigen Verantwortlichkeit (Single-Responsibility-Prinzip), das Prinzip der Offen- und Verschlussenheit (Open-Closed-Prinzip), das Liskovsche Substitutionsprinzip (Liskov-Substitution-Prinzip), das Schnittstellenaufteilungsprinzip (Interface-Segregation-Prinzip) und das Abhängigkeits-Umkehr-Prinzip (Dependency-Inversion-Prinzip), bekannt sind und erstmals von [93] eingeführt wurden. Basierend auf den obigen Ausführungen empfiehlt es sich für Softwarearchitekturen in den meisten Fällen, eine lose Kopplung und eine hohe (funktionale) Kohäsion der Komponenten anzustreben.

Um der zunehmenden Komplexität von Systemen gerecht zu werden, hat sich das Konzept der Layer (Schichten) etabliert, dass die Abstraktion und Trennung von Teilaspekten einer Software, wie z. B. der Datenverarbeitung oder der Benutzerinteraktion, ermöglicht.

Einige Ansätze, die in jüngster Zeit im Kontext von CPS besonders populär sind, sind die Multi-agentensysteme (MAS), die zur Bereitstellung verteilter Intelligenz genutzt werden, und das Paradigma der service-orientierten Architekturen (SOAs), die es ermöglichen, die Idee des Plug & Produce umzusetzen [94, 95]. Darüber hinaus beinhaltet das von IBM eingeführte MAPE (oder MAPE-K) Referenzmodell [96] die wesentlichen Funktionen der architekturbasierten Selbstadaptation, d.h. sogenannte Monitor-, Analyze-, Plan-, Execute- und Knowledge-Komponenten (also Komponenten für das Überwachen, Analysieren, Planen, Ausführen und für das Wissen), um mit Ungewissheiten bzw. Dynamik während der Laufzeit umzugehen [97]. Ein Grundgedanke dieses Konzepts ist die bereits erwähnte *Trennung von Verantwortlichkeiten* (Separation of Concerns).

Das **Referenzarchitekturmodell Industrie 4.0** (RAMI 4.0) [98] wurde entwickelt, um die wesentlichen Elemente des Industrie 4.0-Ansatzes zusammenzuführen. Dazu bildet es ein dreidimensionales Schichtenmodell ab, das aus den drei Achsen *Hierarchy Levels* (Hierarchie), *Life Cycle & Value Stream* (Lebenszyklus & Wertschöpfung) und *Layers* (Architektur) besteht. Die *Layer-Achse* ist wiederum in die sechs Layer: *Business* (Geschäftsprozesse), *Functional* (Funktionen), *Information* (Informationen), *Communication* (Kommunikation), *Integration* (Integration), und *Asset* (Asset („Gegenstand“)) unterteilt. Da RAMI 4.0 verschiedene Nutzerperspektiven einbezieht, bietet es ein gemeinsames Verständnis für Standards und Anwendungsfälle [98]. RAMI 4.0 erhebt jedoch nicht den Anspruch, die Architektur per se darzustellen, sondern lediglich den groben Rahmen für CPPS im Kontext von Industrie 4.0. Dabei legt RAMI 4.0 die Verwendung von Industrie 4.0-Komponenten zugrunde, also von kommunikationsfähigen Komponenten, die als digitale Entitäten verwaltet werden, z.B. mittels einer Verwaltungsschale (VWS) [99]. Eine detailliertere Betrachtung der VWS findet sich in [100], die auch einen automatisierten Entwurf und die Integration von VWS für I4.0-Komponenten behandelt.

Basierend auf einer Analyse von Technologie-Stacks hinsichtlich der Anwendung von Adaptionsmechanismen in der Literatur haben die Autoren von [97] eine grundlegende Multi-Layer-Architektur für CPS abgeleitet. Diese besteht aus sechs separaten Layern, die wie folgt bezeichnet werden: *Physical Layer*, *Proxy Layer*, *Communication Layer*, *Service and Middleware Layer*, *Application Layer*, und als optionale Erweiterung das *Social Layer*. Des Weiteren haben die Autoren von [97] in der Literatur den Trend beobachtet, verschiedene Adaptionsmechanismen zu kombinieren, die über mehrere Layer des Technologie-Stacks hinweg miteinander interagieren und sich koordinieren. Prominente Beispiele für diese Kombinationen sind laut [97] *Smart Elements & MAPE*, *MAS & Smart Elements* oder *MAS & MAPE*. Wobei *Smart Elements* physische Komponenten mit der Fähigkeit zur Selbstadaption darstellen.

Die Arbeit von [101] zielte darauf ab, die Ergebnisse einiger früherer Forschungsarbeiten zu konsolidieren, um eine Referenzarchitektur für flexible CPPS mit Plug & Produce-Fähigkeiten, die sogenannte PERFoRM-Architektur, abzuleiten. Ein Teil dieser Arbeiten stammt dabei aus den Forschungsprojekten IDEAS [102], SOCRADES [103], GRACE [104] und IMC-AESOP [105]. Die PERFoRM-Architektur [106] besteht aus drei Layern und nutzt das SOA-Paradigma. Wobei laut DIN SPEC 16593-1, die sich mit SOAs im Kontext von RAMI 4.0 befasst, SOAs einen der technologischen Eckpfeiler der Industrie 4.0-Vision bilden. Die SOA-basierte Fertigungs-Middleware stellt das Kernelement der PERFoRM-Architektur dar, um eine nahtlose vertikale und horizontale Integration verschiedener Komponenten (z. B. Tools oder Produktionskomponenten) zu ermöglichen. Diese Architektur ist eines von vielen Beispielen, die eine Middleware einsetzen, ein Ansatz, der die in RAMI 4.0 vorgeschlagenen Kommunikations- und Informationslayer abdeckt [101]. Neben der Middleware sind standardisierte Schnittstellen sowie Technologie-Adapter, die zusammen eine Wrapper-Funktionalität (z. B. für die Integration von Legacy-Komponenten) realisieren, wesentliche Merkmale der PERFoRM-Architektur, um Heterogenitätsprobleme durch die Realisierung einer *gemeinsamen Sprache* innerhalb des Systems zu überwinden.

Die vorstehenden Ausführungen stellen lediglich einige prominente Beispiele von Referenzarchitekturen und -mustern dar. Weitere Ansätze, wie etwa die 5C-Architektur, die einen Leitfaden für die Implementierung von CPS darstellt [107], sind in der Literatur zu finden. Als weitere prominente Referenzarchitekturen für das Internet der Dinge können die Industrial Internet Reference Architecture (IIRA) und die Internet of Things - Architecture (IoT-A) genannt werden, die in [108] erörtert werden.

## Diskussion

Obwohl ihr Wert unbestritten ist, sind die genannten Referenzarchitekturen und -muster naturgemäß auf einer eher abstrakten Ebene und können in der Regel nicht unmittelbar und ohne weiteren Aufwand im Hinblick auf den spezifischen Anwendungsfall eingesetzt werden.

Viele modernere Architekturen setzen SOAs oder MASs ein, um eine hohe Flexibilität und Rekonfigurierbarkeit zu erreichen und um mit Heterogenität (z.B. aufgrund unterschiedlicher Komponentenhersteller) umzugehen, indem sie eine lose Kopplung verteilter gekapselter Funktionalität realisieren. Hierzu sei erwähnt, dass sich MASs, die rein reaktiv realisiert werden, nach außen hin nicht von SOAs unterscheiden [95]. Zudem ist in der Literatur ein allgemeiner Trend zur Nutzung von Middleware zu erkennen.

*Auf der Grundlage der obigen Ausführungen und in Abhängigkeit des betrachteten Anwendungsfalls und seiner Anforderungen kann eine der beiden folgenden Optionen gewählt werden: Die erste Möglichkeit besteht darin, eine einzelne Referenzarchitektur oder ein Muster zu konkretisieren, während die zweite Möglichkeit darin besteht, eine eigene Architektur zu entwickeln, die von mehreren bestehenden Referenzarchitekturen und Mustern inspiriert ist. Letzteres wurde gewählt, um die in Kapitel 3.4 vorgestellte CPPS-Architektur zu erhalten, die das selbstorganisierte Rekonfigurationsmanagement ermöglicht und dabei eine hohe Rekonfigurierbarkeit und Flexibilität bietet.*

## 2.3 Wissensmodellierung und Wissensverwaltung

Um die Realisierung von verschiedensten Methodiken innerhalb von CPPS, oder CPS im Allgemeinen, zu ermöglichen, ist ein Konzept zur Wissensmodellierung und -verwaltung eine unerlässliche Voraussetzung, um das benötigte Wissen in geeigneter Form zur Verfügung stellen zu können. Dieses Konzept kann dabei nicht völlig losgelöst von der Systemarchitektur betrachtet werden, da es in den Architekturansatz integriert werden muss. Das bedeutet, dass das notwendige Wissen vorgehalten und den verteilten Komponenten des Systems so zur Verfügung gestellt werden muss, dass diese darauf aufbauend ihre jeweilige Funktionalität umsetzen können. Grundlage für die vertikale wie auch die horizontale Integration über alle Layer und Komponenten eines Produktionssystems hinweg ist dabei ein einheitliches Informationsmodell [109, 110]. Dieses ist erforderlich, da CPPS (oder CPS im Allgemeinen) häufig aus Komponenten bestehen, die von unterschiedlichen Herstellern stammen, wodurch eine semantische Heterogenität entsteht [111]. Der Begriff semantische Heterogenität bezeichnet dabei das Vorhandensein verschiedener Möglichkeiten, um unterschiedliche oder auch äquivalente Konzepte auszudrücken [112]. Ein weiterer Treiber der Heterogenität ist die bereits erwähnte zunehmende Häufigkeit von Änderungen in Produktionssystemen. Neben semantischer Heterogenität sind in diesem Zusammenhang vor allem syntaktische und strukturelle Heterogenitäten zu berücksichtigen [113].

Jüngere Ansätze wie [111] zielen daher auf eine systemweite semantische Interoperabilität und somit auf einen Daten- und Informationsaustausch, der durch eine eindeutige, gemeinsame Bedeutung gekennzeichnet ist, ab. Um dies zu erreichen, ist die bereits erwähnte Verwendung eines

gemeinsamen Informationsmodells ein weit verbreiteter Ansatz und findet sich in diversen Referenzarchitekturen wieder [114]. Es gibt jedoch mehrere Möglichkeiten, ein solches gemeinsames Informationsmodell zu erreichen und aufrechtzuerhalten, insbesondere wenn ein System die Entwurfsphase hinter sich gelassen hat.

Darüber hinaus können Informationsmodelle entweder auf den jeweiligen Anwendungsfall zugeschnitten sein oder aus Gründen der Wiederverwendbarkeit und Effizienz von einer Gruppe von Stakeholdern standardisiert werden [110, 115]. Ein prominentes Beispiel für standardisierte Informationsmodelle sind die in OPC UA<sup>4</sup> eingesetzten "Companion Specifications" [116]. Ein Ansatz für ein allgemeines Maschinenmodell für CNC-Werkzeugmaschinen auf der Grundlage von OPC UA wird in [117] vorgestellt, während ein allgemeines OPC UA-Modell für CPS in [118] vorgeschlagen wird. Des Weiteren gibt es eine wachsende Anzahl von standardisierten Informationsmodellen in der VWS-Community [119]. Hier werden Informationen und Daten auf standardisierte Weise mit Hilfe von so genannten Teilmodellen beschrieben, die man sich als digitale Vorlagen vorstellen kann und die durch den Einbezug von Standards wie ECLASS zur Sicherstellung der Interoperabilität aufgewertet werden [116]. Dabei ist die Definition von weiteren Teilmodellen Gegenstand aktueller Arbeiten in verschiedenen Normungsgremien. Dabei steht das Konzept der VWS nicht im Widerspruch zum Konzept der MAS. Vielmehr kann das MAS-Konzept genutzt werden, um eine VWS zu implementieren [120].

Es gibt zwei bedeutende Disziplinen, die sich mit dem erwähnten Problem der Erstellung und Aufrechterhaltung eines gemeinsamen Informationsmodells befassen. Die eine ist das etablierte Software-Engineering und die andere das ontologische Engineering, das in den letzten Jahren stark an Aufmerksamkeit gewonnen hat. In ähnlicher Weise unterscheiden die Autoren von [115] zwischen zwei Kategorien von Ansätzen für die Modellierung von Fähigkeiten: eine Kategorie verwendet Ontologien als formale Modelle, während die andere XML-basierte Modelle verwendet. Dabei ist die Fähigkeitsmodellierung ein wichtiger Teil der Modellierung für das Rekonfigurationsmanagement, da es notwendig ist, die Anforderungen an die Produkte mit den Fähigkeiten des CPPS vergleichen zu können [31].

Die Tatsache, dass Ontologien sowohl informal oder semi-formal spezifiziert werden können, z.B. mit UML-Klassendiagrammen, Entity-Relationship-Modellen und semantischen Netzen, als auch formal spezifiziert werden können, z.B. mit OWL, fördert die Unklarheiten bezüglich des Ontologie-Begriffs und führt schließlich zu einer Vielzahl von Modellen, Konzepten und Spezifikationen, die als Ontologien veröffentlicht werden [121]. Der erstgenannte Typ der Repräsentation eines Informationsmodells wird auch als Lightweight Ontology (LWO) bezeichnet und umfasst

---

<sup>4</sup> Siehe auch: <https://opcfoundation.org/about/opc-technologies/opc-ua/>

Konzepte, deren Taxonomien und Relationen sowie deren Eigenschaften. Im Gegensatz dazu wird der zweite Typ auch als Heavyweight Ontology (HWO) bezeichnet und erweitert den ersten Typ um Axiome und Constraints [111]. Darüber hinaus besitzen die UML-basierten Modelle (z.B. Klassendiagramme), die im Rahmen des Software-Engineerings erstellt werden, eher eine geringere semantische Ausdruckskraft, da sie LWOs darstellen, während das ontologische Engineering klassischerweise darauf abzielt, durch die Entwicklung von HWOs eine hohe semantische Ausdruckskraft zu erreichen. Die Entwicklung einer LWO schließt jedoch eine Weiterentwicklung hin zu einer HWO nicht aus, sondern stellt gleichzeitig einen ersten Schritt dieser Weiterentwicklung dar. Dabei ist es wichtig zu beachten, dass die semantische Ausdruckskraft als eine graduelle und nicht als eine binäre Größe betrachtet werden sollte.

Nach [122] stellen Ontologien eine mögliche Realisierung von Wissensbasen in Anwendungen der künstlichen Intelligenz dar<sup>5</sup>. Während sich eine der meistverwendeten Definitionen von Ontologien aus [123] als "eine formale, explizite Spezifikation einer gemeinsamen Konzeptualisierung" übersetzen lässt, wird im Rahmen dieser Forschungsarbeit die erweiterte Definition von [121] zugrunde gelegt, welche sich als "eine Ontologie ist eine formale, explizite Spezifikation einer gemeinsamen Konzeptualisierung, die sich durch eine hohe semantische Ausdruckskraft auszeichnet, welche für eine erhöhte Komplexität erforderlich ist" übersetzen lässt. Letztere beinhaltet die Vorteile von *vollwertigen* Ontologien und hilft somit, sie von anderen Lösungen abzugrenzen und ist folglich hilfreich für die Diskussion sowie für die Entscheidung für die eine oder die andere Lösung.

Die Autoren von [121] haben eine detaillierte Analyse von Ontologien durchgeführt und erörtern, warum diese bislang nicht so erfolgreich waren, wie sie es hätten sein können, und wie diese Problematik adressiert werden kann. Darüber hinaus wird eine Anleitung zur Beantwortung der Frage gegeben, wann Ontologien eingesetzt werden sollten und wann nicht, welche für diese Forschungsarbeit von besonderem Interesse ist. Zusammengefasst sollen demnach die folgenden drei Schritte durchgeführt werden [121]:

- Die Feststellung, ob die Annahme einer geschlossenen oder offenen Welt angemessen ist<sup>6</sup>,
- die Wahl des Repräsentationsmodells auf der Grundlage des erforderlichen Maßes an Ausdruckskraft,
- und anschließend, basierend auf diesem und dem erforderlichen Grad der gemeinsamen Nutzung, die Wahl der Repräsentationssprache.

---

<sup>5</sup> An dieser Publikation wurde im Rahmen dieser Forschungsarbeit mitgewirkt.

<sup>6</sup> Weiter verbreitet sind hierbei die englischen Begriffe der closed- bzw. open-world assumption.

Daher schlagen sie vor, zunächst zu prüfen, ob eine Ontologie im Vergleich zu einer reinen Taxonomie wirklich zwingend erforderlich ist, da eine Ontologie mit einem erhöhten Aufwand verbunden ist, welcher wiederum von der konkreten Ausprägung der Ontologie abhängt. Allerdings bildet eine Taxonomie selbst das Rückgrat einer Ontologie. Eine Kernaussage hierzu lautet in [121] wie folgt: „Eine ontologiebasierte Entwicklung wird als nicht geeignet angesehen, um homogene Datensichten, Datenintegration und Workflows effizient zu verwirklichen“. Darüber hinaus sind Ontologien in der Regel komplex, und die Identifikation von Ähnlichkeiten zwischen bestehenden Ontologien und den Informationen, die für den spezifischen Anwendungsfall abgebildet werden sollen, ist eine schwierige und mühsame Aufgabe, was die Wiederverwendung von Ontologien behindert und stattdessen zur Entwicklung neuer Ontologien von Grund auf führt [111].

In der Literatur gibt es eine Vielzahl von Ansätzen, die sich wie erwähnt grob in solche, die Software-Engineering verwenden, und solche, die ontologisches Engineering einsetzen, unterteilen lassen. Im Folgenden wird zunächst ein kurzer Auszug von Beispielen gegeben, die auf dem klassischen Software-Engineering beruhen.

In [124] wird eine virtuelle CPPS-Repräsentation als Teil eines Assistenzansatzes für die Rekonfiguration von CPPSs und zur Durchführung eines attributiven Mappings von CPPS-Fähigkeiten mit Produktionsanforderungen für einen interaktiven Mensch-Roboter-Montageprozess verwendet. Diese Repräsentation beinhaltet die CPPS-Struktur im AML-Format, die dann in ein UML-Datenmodell umgewandelt wird, um das Mapping zu ermöglichen und durchzuführen.

Die Autoren von [125] stellen ein integriertes Planungswerkzeug vor, das ein Product-Lifecycle-Management-System und eine Prozesssimulation für die Rekonfigurationsplanung von flexiblen Montagesystemen kombiniert. Sie verwenden ein erweitertes Entity-Relationship-Datenmodell, das ein Produkt-Prozess-Ressource- (PPR)-Teilmodell, ein Simulations- und ein Produktionsprogramm-Teilmodell enthält. Darauf aufbauend bietet das Werkzeug Hilfestellung bei der Generierung des Simulationsmodells und bei der Simulationsdurchführung für verschiedene Planungsalternativen. Schließlich erhält der Anwender Bewertungsparameter, die einen Vergleich der Alternativen ermöglichen.

Im Gegensatz zum gängigen Stand der Wissenschaft und Technik wird in [126] ein Modellierungsansatz für Fertigungsressourcen beschrieben, bei dem Prozesse auf Basis ihrer Input- und Output-Elemente und nicht durch definierte Prozessparameter beschrieben werden. Das PPR-Konzept wird dabei unter Anwendung der Formalisierten Prozessbeschreibung (FPB) auf Basis der VDI/VDE 3682-Richtlinie<sup>7</sup> [127] genutzt.

---

<sup>7</sup> Herausgegeben von der VDI/VDE Gesellschaft für Mess- und Automatisierungstechnik.

Nachfolgend werden einige Beispiele für die Kategorie der Ansätze, die das ontologische Engineering nutzen, angeführt.

In [128] wird ein Ansatz vorgestellt, der zu einem Framework für die Analyse der Machbarkeit in frühen Entwurfsphasen führt. Das Framework ermöglicht projektspezifische Beschreibungen von Ressourcen und Produkten, die auf einer Domänen-Ontologie basieren. Darüber hinaus werden verschiedene Domänen-Ontologien mit einer zwischengeschalteten Engineering-Ontologie abgeglichen, die wiederum an der Top-Level-Ontologie DOLCE ausgerichtet ist, um Wiederverwendbarkeit und generische Abfragen unabhängig von domänenspezifischem Wissen zu ermöglichen.

Die Autoren von [129] skizzieren die Entwicklung der OWL-basierten Manufacturing Resource Capability Ontology (MaRCO), also einer Ontologie für die Fähigkeiten von Fertigungsressourcen. Es wird eine detaillierte Ressourcenbeschreibung bereitgestellt, die im Gegensatz zu anderen eine automatische Inferenz (engl. inference) von kombinierten Fähigkeiten unterstützt. Darüber hinaus beschreiben sie in [130], wie Informationen aus einem Produktmodell und einem Ressourcenmodell verwendet werden, um einen Abgleich zwischen Produktanforderungen und Ressourcenfähigkeiten unter Verwendung eines Prozess-taxonomie-Modells durchzuführen.

Ein Ansatz für ein formales Modell, das sowohl Maschinenfähigkeiten als auch sogenannte ausführbare Skills umfasst, wird in [115] vorgestellt und basiert auf der Verwendung verschiedener Ontology Design Pattern (ODPs), zu Deutsch Ontologie-Entwurfsmuster. Ihr modularer Ansatz zielt darauf ab, die Erweiterbarkeit und Wiederverwendbarkeit zu erhöhen und gleichzeitig den Entwicklungsaufwand zu reduzieren. Dies wird dadurch erreicht, dass jedes ODP auf einem einzigen Industriestandard basiert, z.B. werden abstrakte Fähigkeiten anhand von Prozessen in Übereinstimmung mit der FPB beschrieben. Um letzten Endes eine Ontologie zu erhalten, die Aspekte der verschiedenen Industriestandards abdeckt, wird eine entsprechende Alignment-Ontologie<sup>8</sup> abgeleitet, die schließlich als Fähigkeitsmodell dient.

## Diskussion

Ein Thema, das in der Literatur recht umfassend behandelt wird, ist der Abgleich bzw. das Matchmaking von Produktanforderungen mit den Fähigkeiten eines (cyber-physischen) Produktionssystems bzw. die hierfür notwendige Modellierung, wobei eine Vielzahl unterschiedlicher Ansätze existiert. Die durchgeführte Literaturanalyse hat jedoch gezeigt, dass ein Defizit hinsichtlich des Einbezugs von nicht-funktionalen Aspekten besteht. Somit werden die resultierenden Anwendungen von Methodiken, die nicht-funktionale Kriterien einbeziehen, durch diese Wissensmodellierungs- und verwaltungsansätze nicht unterstützt.

---

<sup>8</sup> Das Ontologie-Alignment befasst sich mit der Abbildung verschiedener Ontologien aufeinander.

Der wesentliche Grund, für diesen speziellen Anwendungsfall keine vollwertige HWO anzustreben, ist der Aufwand, der damit verbunden wäre. Dies gilt insbesondere auch im Hinblick auf die oben hervorgehobene mangelhafte Wiederverwendbarkeit von bestehenden Ontologien, die durch die erwähnte Vernachlässigung von nicht-funktionalen Aspekten zusätzlich erschwert wird. Die Anforderungen, die das selbstorganisierte Rekonfigurationsmanagement an die Wissensrepräsentation stellt, können durch eine Closed-World-Assumption erfüllt werden, d.h. durch die Bereitstellung einer homogenen, konsistenten Sicht auf die (verteilten) Informationen.

Es besteht keine Notwendigkeit ein gemeinsames Verständnis während der Betriebsphase des Produktionssystems auszutauschen, da es eher syntaktische und strukturelle Heterogenitäten aufgrund verschiedener (Produktionsmodul-)Anbieter gibt als eine dynamische Änderung der zugrundeliegenden Semantik. Daher ist es möglich, Interoperabilität und den Umgang mit einer möglichen Hersteller-Heterogenität durch die Verwendung eines gemeinsamen Informationsmodells zu erreichen. Dieses Informationsmodell definiert in erster Linie die Vorlage für die eigentlichen Modelle der Systemkomponenten, z. B. der Produktionsmodule. Dieser Ansatz kann durch die Verwendung von Wrappern erweitert werden, um die Informationen eines diversitären (modulspezifischen) Modellierungsansatzes zu extrahieren und in die systemweite *gemeinsame Sprache* zu übersetzen und so eine größere semantische Heterogenität zu bewältigen. Zudem wollen verschiedene Anbieter nicht immer alle ihre Informationen mittels einer detaillierten Ontologie austauschen, welche klassischerweise zentral zugänglich ist. Daher ist ein minimalistisches gemeinsames Informationsmodell besser geeignet, während weitere, nicht gemeinsam genutzte Informationen enthalten sein können (innerhalb desselben oder auch zusätzlicher Modelle), welche dann z. B. (lediglich) auf Modulebene verwendet werden können.

Natürlich könnte dies auch durch den Einsatz einer vollwertigen Ontologie realisiert werden. Da jedoch das geforderte minimalistische Informationsmodell im Widerspruch zu einer hohen semantischen Ausdruckskraft steht, einem dann ungenutzten wesentlichen Vorteil der Ontologie, ist der damit verbundene Aufwand nicht gerechtfertigt.

Softwareagenten stellen eine geeignete Technologie dar, um die verschiedenen Modelle einzubinden und die oben erwähnte Wrapper-Funktionalität zu realisieren. Ontologien bieten die Möglichkeit für zusätzliche Managementfunktionen durch Abfragen und Schlussfolgerungen<sup>9</sup>, die die Möglichkeiten eines reinen UML/XML-basierten Wissensmodellierungsansatzes übersteigen. Aber auch im Hinblick auf diese Problematik, kann die Softwareagenten-Technologie eingesetzt werden, um den UML/XML-basierten Ansatz diesbezüglich zu bereichern.

---

<sup>9</sup> Weiter verbreitet sind die englischen Begriffe Queries (Abfragen) und Reasoning (Schlussfolgern).



Aus technischer Sicht werden die nachfolgend angeführten acht Vorteile in [121] genannt und beschrieben. Um in Bezug auf den gegebenen Anwendungsfall zu entscheiden, kann weiter untersucht werden, welche dieser Vorteile bereits durch einen UML/XML-basierten Ansatz, der eine LWO repräsentiert, erschlossen werden können und welche erst durch eine HWO bereitgestellt werden. Dabei kann eine LWO die Aspekte *Kommunikation*, *Wissensorganisation*, *Wiederverwendbarkeit*, *Standardisierung* und *Identifikation* sowie eine *T-Box/A-Box-Trennung* zumindest bis zu einem gewissen Grad abdecken. Eine HWO kann darüber hinaus *Inferenz* ermöglichen und die Aspekte der *konstanten Evolution* verbessern.

Im betrachteten Anwendungsfall des Rekonfigurationsmanagements ist es jedoch nicht notwendig, durch Schlussfolgerungen neues Wissen abzuleiten oder eine konstante Evolution in Form eines agilen Schema-Managements während der Betriebsphase zu ermöglichen.

Zudem lässt sich sagen, dass der Stand der Wissenschaft und Technik der Modellierung sowohl die Rekonfiguration als auch den Produktionsaufwand basierend auf nicht-funktionalen Kriterien nicht hinreichend berücksichtigt.

*Auf der Grundlage der obigen Ausführungen wurde entschieden einen neuartigen UML/XML-basierten Ansatz zu verfolgen, da dieser die Anforderungen des Anwendungsfalls „Rekonfigurationsmanagement“ erfüllen und darüber hinaus eine LWO bereitstellt. Darüber hinaus deckt diese LWO die ersten Schritte von Entwicklungsmethoden für Ontologien ab, wie sie beispielsweise in [111] vorgeschlagen werden.*

## 2.4 Rekonfigurationsmanagement

In Kapitel 2.4 erfolgt eine Betrachtung der Literatur im Hinblick auf die methodische Umsetzung der Bestandteile des Rekonfigurationsmanagements im Rahmen von Kapitel 2.4.1 Ermittlung des Rekonfigurationsbedarfs, Kapitel 2.4.2 Generierung alternativer Konfigurationen und Kapitel 2.4.3 Bewertung und Auswahl von Konfigurationen.

### 2.4.1 Ermittlung des Rekonfigurationsbedarfs

Für die Ermittlung des Rekonfigurationsbedarfs und die Generierung alternativer Konfigurationen ist es zunächst erforderlich, die Anforderungen der Produkte mit den Fähigkeiten des CPPS vergleichen zu können, wofür bestehende Ansätze untersucht werden. Die meisten Ansätze in der Literatur stützen sich auf das Produkt-Prozess-Ressource-Konzept (PPR-Konzept). Dieses kann eingesetzt werden, um den Zusammenhang zwischen den drei Elementen herzustellen, also welche Prozesse für ein Produkt benötigt werden und welche Ressourcen diese Prozesse realisieren können und somit für die Herstellung eines Produktes eingesetzt werden können. Dabei gibt es jedoch eine Vielzahl von Variationen. Diese Variationen treten sowohl hinsichtlich der Inhalte,

die modelliert werden, als auch hinsichtlich des gewählten Modellierungsansatzes (vgl. Kapitel 2.3) auf.

Im Hinblick auf die Ermittlung des Rekonfigurationsbedarfs listet Tabelle 2.2 verschiedene Trigger für die Rekonfiguration auf, die als interne (internal) oder externe (external) Trigger kategorisiert werden und bereits in [131] im Rahmen dieser Forschungsarbeit veröffentlicht wurden. Diese Auflistung erhebt jedoch keinen Anspruch auf Vollständigkeit, insbesondere da aufgrund der anhaltenden Volatilität der Anforderungen in der industriellen Automatisierung neue Trigger entstehen können und werden. Während interne Trigger, wie der Ausfall einer Komponente, aus dem Produktionssystem selbst herrühren, haben externe Trigger, wie etwa ein neuer Produktionsauftrag, ihren Ursprung außerhalb der Systemgrenze des Produktionssystems. Eine alternative Benennung der beiden Trigger-Kategorien als betriebsbedingt (operational) und strategisch (strategic) wird in [132] vorgeschlagen. Diese alternative Benennung hat ihren Ursprung in der Art und Weise wie diese Trigger adressiert werden können. So schlägt [133] vor, interne Trigger, falls möglich über die vorhandene Flexibilität der aktuellen Systemkonfiguration zu adressieren, bevor eine Rekonfiguration in Betracht gezogen wird, was einer betriebsbedingten Entscheidung entspricht [134]. Im Gegensatz dazu seien externe Trigger seltener über den Flexibilitätskorridor einer Systemkonfiguration abgedeckt [135] und der Zeithorizont ist im Vergleich zu internen Triggern eher unkritisch, was einer strategischen Entscheidung entspricht [134].

**Tabelle 2.2: Interne und externe Trigger für die Rekonfiguration [131]**

| <b>Interne Trigger</b>                                                 | <b>Externe Trigger</b>                                                         |
|------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| Eine Systemkomponente hat einen Defekt.                                | Ein neues Produkt oder eine neue Produktvariante wird gefordert.               |
| Instandhaltungsmaßnahmen werden notwendig.                             | Ein geändertes Produktionsvolumen wird gefordert.                              |
| Die angewandte Konfiguration erreicht die erwarteten Kennzahlen nicht. | Es wird eine erhöhte Flexibilität des Produktionssystems gefordert.            |
|                                                                        | Eine Verbesserung der Qualitätsmerkmale des Produktionssystems wird gefordert. |
|                                                                        | Die Rechtslage ändert sich (z.B. COVID-19 fordert mehr Raum pro Person).       |
|                                                                        | Die Umgebungsbedingungen des Produktionssystems ändern sich.                   |
|                                                                        | Die Verfügbarkeit des benötigten Produktionsmaterials ändert sich.             |

Im Kontext einer dynamischen Software-Rekonfiguration von Fertigungsservices wird die Ermittlung von Rekonfigurationsbedarf in [86] im Sinne der Entdeckung von Möglichkeiten zur Verbesserung der Produktionseffizienz adressiert. Dazu wird ein WtR-Modell (When to Reconfigure-Modell) verwendet, um basierend auf einer kontinuierlichen Überwachung und Analyse von Daten eine Möglichkeit für eine Software-Rekonfiguration zu identifizieren, wie z.B. die Verschlechterung der Qualität eines Services oder ein Produktwechsel. Im WtR-Modell werden die drei Trigger-Typen ereignisabhängig, periodisch und trendabhängig unterschieden. Ereignisse beziehen sich hierbei auf Änderungen des Servicezustandes, wie z.B. durch einen Ressourcenausfall oder das Entfernen einer Ressource oder eines Services. Darüber hinaus findet eine periodische Überprüfung der aktuellen Service-Performance statt. Eine Besonderheit dieses Ansatzes stellt die Identifikation von Trends dar, die auf Basis von Methoden des maschinellen Lernens (z.B. Anomalie-Detektion, Regressionsmodellen oder Cluster-Analyse) frühzeitig potenzielle Probleme, also Möglichkeiten für eine Service-Rekonfiguration mit dem Ziel der Prävention von Performance-Einbußen, erkennt.

In [77] werden Skills im Sinne einer generalisierten Beschreibung der Fähigkeiten einer Komponente anstelle von proprietärem Programmcode verwendet, um einen Abgleich von Produktionsanforderungen und den Fähigkeiten des Produktionssystems zu ermöglichen. Die Verwendung solcher Skills ist in der Literatur weit verbreitet und ermöglicht eine herstellerübergreifende Interoperabilität von Automatisierungskomponenten. Die in [77] eingesetzte Datenanalyse-Komponente nutzt einen DBSCAN-Algorithmus zur Clusteranalyse um signifikante Datengruppen in einem mehrdimensionalen Raum identifizieren zu können. Nicht konvexe Hüllen werden für eine Anomalie-Detektion eingesetzt, um zwischen normalen und abnormalen Daten zu unterscheiden, was in einer symbolischen Repräsentation des CPPS-Status in Form von Alarmen resultiert. In der Alarm-Management-Komponente werden schließlich die ursächlichen Alarme identifiziert und basierend auf Domänenwissen, das mittels OWL in einer Ontologie abgebildet wird, einem Modul zugeordnet das dann als fehlerhaft identifiziert wird. Hinsichtlich der Integration externer Trigger werden auf Basis von Produktionsaufträgen Pläne generiert, die aus einer Sequenz von Skills bestehen und somit den skillbasierten Abgleich der Produktionsanforderungen mit den Fähigkeiten des Produktionssystems zu ermöglichen.

Ansätze wie [76] nutzen neben in OWL beschriebenen Ontologien zusätzlich SWRL um komplexere Relationen abbilden zu können und dadurch ein komplexeres Reasoning zu erlauben. Konkret wird in [76] die Manufacturing's Semantics Ontology (MASON-Ontologie), bereichert um SWRL-Regeln, eingesetzt um den Abgleich der Produktionsanforderungen (hier Medikamentenbestellungen) mit den Fähigkeiten des Produktionssystems vergleichen zu können. Immer wenn eine neue Medikamentenbestellung eingeht, wird diese zunächst in der Cloud entgegengenommen, analysiert und die Informationen in das XML-Format umgewandelt, um sie weiter verarbeiten zu können. Im nächsten Schritt nimmt ein in der Cloud laufender Prozess die XML-Daten

entgegen und wandelt sie in eine SPARQL-Abfrage um. Anschließend werden die Statusdaten der pharmazeutischen Produktionsressourcen in der Datenbank aktualisiert. Schließlich fragt der Reasoner die Wissensdatenbank mit SPARQL ab. Das Ergebnis der Abfrage enthält Informationen darüber, welche Ressourcen zur Verfügung stehen und für die spezifischen Produktionsanforderungen ausgewählt wurden (und auch darüber, wie diese Ressourcen in einer Produktionssequenz zusammenarbeiten). Bezüglich der Beachtung eines internen Triggers wird ein „condition monitoring“ erwähnt und das Betriebsausfälle in die Cloud gemeldet werden, worauf basierend es einfach möglich ist die Ermittlung von Rekonfigurationsbedarf zu triggern, auch wenn dies in [76] nicht explizit beschrieben ist.

Bei der Betrachtung von externen Triggern dominiert der Eingang eines neuen Produktionsauftrages, so wird etwa in [84], am Beispiel der Produktion von Flip-Flop-Sandalen, ein Rekonfigurationsbedarf in Abhängigkeit des Durchsatzes identifiziert. Ist der Durchsatz der aktuellen Konfiguration des Produktionssystems geringer als durch den eingegangenen Produktionsauftrag gefordert, so wurde ein Rekonfigurationsbedarf ermittelt (und zusätzliche Produktionsressourcen können subsequent aktiviert werden). Ebenso wird ein Rekonfigurationsbedarf identifiziert, wenn der neue Produktionsauftrag einen Durchsatz fordert, der geringer ist als der durch die aktuelle Konfiguration des Produktionssystems mögliche Durchsatz (woraufhin subsequent Produktionsressourcen deaktiviert werden können).

Ebenso wird in [87], im Sinne eines verhandlungsbasierten freien Marktwirtschaft-Ansatzes, auf neue Produktionsaufträge derart reagiert, dass untersucht wird ob die Rekonfiguration einer Ressource profitabel wäre. Dazu werden Produktionsaufträge dynamisch zusammengruppiert, sofern sie die gleichen Ressourcenfähigkeiten benötigen. Der Nutzen für eine Ressource, der sich aus der Umstellung auf eine andere Produktcharge ergibt, wird mit den Marktbedingungen verglichen, um eine wirtschaftliche Bewertung der Gesamtbedingungen zu erhalten. Die Kosten einer solchen Ressourcenrekonfiguration werden als äquivalente entgangene Betriebseinnahmen berechnet und den Aufträgen, die eine Umstellung erfordern, in Rechnung gestellt. Dazu werden Produktionsaufträge mit einer Pseudo-Währung initialisiert wodurch eine Verhandlung mit den Produktionsressourcen, welche die Kosten für Produktionsprozesse basierend auf Auslastung, Warteschlangen und Rekonfigurationskosten bestimmen ermöglicht wird. Für diese Verhandlung wird das „contract net“ Protokoll verwendet.

In Bezug auf interne Trigger wird häufig der Fall adressiert, dass die zuvor vorhandene Funktionalität nicht mehr gewährleistet ist. Dazu wird in [78] ein Systemmodell mit Hilfe der Aussagenlogik (engl. Propositional Logic) aufgebaut und zur Ermittlung von Rekonfigurationsbedarf überprüft, ob mindestens eine Zustandsvariable  $x$  ihren gültigen Bereich verlassen hat. Dazu werden

die gültigen Bereiche jeder Zustandsvariable zuvor durch einen oberen und einen unteren Grenzwert spezifiziert. Im beschriebenen Anwendungsfall aus der Verfahrenstechnik, dem Tennessee Eastman Process [136], stellt beispielsweise die Temperatur eine solche Zustandsvariable dar.

In ähnlicher Weise zielt [88] darauf ab, den Zustand der einzelnen Arbeitsstationen aufrechtzuerhalten. Dazu werden der Status der Ressourcen, der Fortschritt der Produktionsprozesse und die Ausgabe von Warnungen zu Prozess-, System- und Ressourcenfehlern berücksichtigt. So werden Ereignisse, wie z.B. das Verlassen einer Arbeitsstation durch einen Werker, vom Planungs- und Scheduling-Tool erfasst und durch eine sogenannte Context-Engine, hier als Nichtverfügbarkeit einer Ressource, interpretiert und ein potenzieller Rekonfigurationsbedarf ermittelt.

Die in [91] vorgestellte Arbeit befasst sich mit einem verteilten Digitaler Zwilling-Framework (DZ-Framework), das sowohl lokale als auch globale DZs und Entscheidungsfindung umfasst. Die Arbeit konzentriert sich mehr auf diese grundlegende DZ-Struktur und insbesondere auf einen Ansatz zur Erkennung möglicher Fehler auf der Grundlage eines adaptiven Schwellenwertverfahrens als auf das Thema Rekonfiguration. Die Detektion von anormalem Verhalten kann jedoch zur Ermittlung von Rekonfigurationsbedarf verwendet werden, und ein Rekonfigurationsmanagement könnte in dieses eher allgemeine und abstrakte DZ-Framework eingebettet werden.

## 2.4.2 Generierung alternativer Konfigurationen

Wie bereits im vorherigen Teilkapitel erwähnt, ist es auch für die Generierung alternativer Konfigurationen von Nöten die Produktionsanforderungen mit den Fähigkeiten des Produktionssystems abgleichen zu können. An dieser Stelle liegt der Fokus jedoch in der Identifikation von geeigneten Alternativen, auf der Maschinen- oder Systemebene, welche die Produktionsanforderungen komplett oder zumindest teilweise erfüllen können. Dabei kommen in der Literatur zum Teil verschiedene Formen der Optimierung zur gezielten Generierung von Konfigurationsalternativen für eine anschließende Bewertung und Auswahl zum Einsatz.

Bei der Generierung alternativer Konfigurationen lässt sich zunächst differenzieren ob und in welcher Form die Hardware und Software auf der Maschinenebene miteinbezogen werden.

Im Hinblick auf den Software-Aspekt wird z.B. in [76] ein Ansatz vorgestellt, um unterschiedliche Funktionalitäten eines invarianten physischen Systems (Hardware) nutzen zu können. Dabei wird die IEC 61499 für die Modellierung der Funktionalität sowie für die Maschinensteuerung mittels Funktionsbausteinen verwendet. Somit wird die durch die alternativen Software-Komponenten ermöglichte Funktionalität nach oben gekapselt angeboten und mit Hilfe der bereits in Kapitel 2.4.1 beschriebenen Ontologie wird die Generierung alternativer Konfigurationen ermöglicht.

Ein weiterer Ansatz, der sich im Zuge einer Rekonfiguration von Fertigungsservices auf den Software-Aspekt fokussiert ist in [86] beschrieben und zielt auf die Produktivitätssteigerung ab. Hierzu kennt ein Softwareagent, welcher eine Produktionsressource verwaltet, einen Katalog an verfügbaren alternativen Fertigungsservices. Diese Fertigungsservices sind dabei hinsichtlich ihrer Verfügbarkeit, Ausführungszeit und ihres Durchsatzes beschrieben, sodass diese Informationen im weiteren Verlauf verwendet werden können.

Zahlreiche Ansätze, wie z.B. [79–83, 87, 89] betrachten die Rekonfiguration aus einer klassischen, maschinenbaulichen Sicht, dies zeigt sich insbesondere auf der Maschinenebene. Dabei wird der Software-Aspekt, bzw. das Vorhandensein von Software-Komponenten oft nicht explizit berücksichtigt und der Fokus auf die Hardware gelegt bzw. es wird vereinfacht in abstrakten Funktionsmodulen gedacht.

Weiterhin basieren viele Ansätze, wie z.B. [79, 80, 82, 89], auf dem Einsatz von sogenannten *Reconfigurable Machine Tools* (RMTs) nach [13], die einen Kernbestandteil von RMSs darstellen [137]. RMTs können dabei in verschiedene Konfigurationen überführt werden und bieten somit anpassbare Funktionalität und Kapazität, indem ihre Basismodule erhalten bleiben, während das Hinzufügen, Entfernen, oder Modifizieren von Hilfsmodulen möglich ist [12, 49, 79]. In [80] werden Basismodule als strukturelle Elemente beschrieben, die permanent mit dem RMT verbunden sind, während Hilfsmodule kinematische oder bewegungsgebende Module, wie z. B. Spindeln, sind, die montiert oder demontiert werden können, um unterschiedliche operative Fähigkeiten bereitzustellen.

So werden z.B. in [79] und [82] verschiedene Konfigurationsalternativen auf der Maschinenebene berücksichtigt, welche jeweils über ein Set aus Basis- und Hilfsmodulen verfügen und dadurch eine variable Menge an Operationen verfügen. Weit verbreitete Angaben in Bezug auf die Konfigurationsalternativen sind beispielsweise die Montage- bzw. Demontagezeiten der Hilfsmodule von RMTs, sowie die Ausführungszeiten der jeweils angebotenen Operationen, welche typischer Weise als feste Werte angenommen werden und etwa in [80] verwendet werden. In [80] und [81] werden die Konfigurationsalternativen lediglich anhand der Hilfsmodule und der möglichen Operationen unterschieden und auf die Betrachtung der, fest verbauten, Basismodule verzichtet. Hierzu ist außerdem anzumerken, dass die Beschreibung der Konfigurationsalternativen in selteneren Fällen, wie z.B. in [81], zudem auch die Tool Approach Directions (TADs), bzw. den Tool Orientation Space (TOS) [82], umfasst. Demgegenüber wird in [89] ein Beispiel für eine abstrakte Betrachtung der Maschinenebene beschrieben. Dabei werden lediglich die vorhandenen Konfigurationsalternativen unterschieden, ohne die Komponentenebene (etwa über Basis- und Hilfsmodule) zu betrachten. Informationen dieser Art werden oft, wie z.B. in [79–82, 89] zentral (in Tabellenform) vorgehalten und als gegeben vorausgesetzt.

In [78] wird ein Ansatz präsentiert bei dem das in Kapitel 2.4.1 beschriebene, mit Hilfe der Aussagenlogik aufgebaute qualitative Systemmodell des CPPS nutzt, um abzuschätzen wie sich das System entwickelt bzw. verhalten wird. Hierzu wird auch die Steuerung und dessen Auswirkung auf das System modelliert. Das Systemverhalten wird basierend auf apriori-Wissen abgebildet indem Kausale Graphen die Zusammenhänge zwischen Eingangs- und Zustandsvariablen beschreiben. Anschließend wird das Systemmodell durch einen SAT-Solver gelöst der lediglich darauf abzielt ein Minimum an Eingangsvariablen, die beispielsweise den Austausch von Komponenten oder die Verwendung redundanter Hardware, oder auch das aktivieren/deaktivieren eines Heizelements integrieren, zu verändern und dennoch eine valide Konfiguration zu erreichen. Das Ergebnis ist eine Eingabemaske, die den Eingangsvariablen des hybriden Automaten feste Werte zuweist und die Wiederherstellung einer validen Konfiguration gewährleistet.

Auf der Systemebene hingegen gibt es unterschiedliche Ansätze die Maschinenauswahl, die Positionierung dieser Maschinen im Produktionslayout, sowie die Anpassung der Produktionssequenz zu adressieren.

Viele Ansätze in der Literatur, wie z.B. [79–81, 89], stellen ein mathematisches Modell auf, um gezielt Lösungen auf der Systemebene generieren zu können, diese unterscheiden sich jedoch stark in der Art und Weise, wie das zugrundeliegende Problem mathematisch abgebildet wird und auch in der konkreten Ausprägung des zugrundeliegenden Problems. Beispielsweise wird in [89] davon ausgegangen, dass für jede benötigte Operation mindestens eine geeignete RMT-Konfiguration existiert und dass alle Produktionsaufträge, die eingeplant werden sollen, bereits initial bekannt sind. Insbesondere kommen unterschiedliche Arten der Optimierung zum Einsatz, wobei sowohl Ansätze zur einkriteriellen [80, 83, 89] als auch zur multikriteriellen Optimierung [79, 81] existieren. Darüber hinaus unterscheiden sich die Ansätze in der Auswahl der betrachteten Kriterien und auch in der Art und Weise, wie diese Kriterien jeweils zusammengesetzt sind.

So wird in [79] eine multikriterielle Optimierung für die Konfigurationsauswahl einer mehrstufigen Produktionslinie vorgestellt, die die monetären Produktionskosten minimieren und gleichzeitig die Rekonfigurierbarkeit, die Flexibilität und die Zuverlässigkeit maximieren soll. Die Rekonfigurierbarkeit eines jeden RMT wird dabei über die Anzahl der vorhandenen alternativen RMT-Konfigurationen im Verhältnis zu den Rekonfigurationsaufwänden, die sich aus der erforderlichen Anzahl an hinzuzufügenden, zu entfernenden und anzupassenden Hilfsmodulen ergeben, ermittelt. Dazu wird die jeweilige Anzahl mit einer spezifischen Gewichtung für das Hinzufügen, Entfernen und Anpassen der Hilfsmodule multipliziert. Die Flexibilität hingegen wird anhand der Anzahl der möglichen Operationen quantifiziert, die eine RMT-Konfiguration durchführen kann. Die Zuverlässigkeit wird durch eine simple Multiplikation der Zuverlässigkeitswerte der vorgesehenen RMT-Konfigurationen berechnet, da in der betrachteten mehrstufigen Produktionslinie jedes RMT die stufenspezifische Operation durchführt. Immer dann, wenn konkurrierende Optimierungskriterien miteinander in Konflikt stehen, ist es unmöglich, eine Lösung zu finden, die für

alle Kriterien das Optimum darstellt. In diesem Fall ist es in der Literatur gängige Praxis, die Pareto-Front zu approximieren, d.h. die nicht-dominierten Lösungen zu finden. In [79] wird dazu ein *Non-dominated Sorting Genetic Algorithm II* (NSGA-II)<sup>10</sup> verwendet. In diesem Ansatz ist neben der reinen Auswahl der RMTs (Maschinen) auch die Auswahl der Konfigurationen und deren Allokation zu einer Produktionsstufe auf Basis einer vorgegebenen Bill of Process (BOP) integriert.

In [89] werden verschiedene Ansätze zur Lösung eines Produktionsplanungsproblems in einer Produktionsstätte mit RMTs vorgestellt. Dabei werden zwei MILP-Modelle definiert, von denen eines einen *positionsbasierten* und das andere einen *sequenzbasierten* Ansatz verfolgt. Der positionsbasierte Ansatz weist dabei Operationen zu vordefinierten *Zeit-Positionen* zu, wohingegen der sequenzbasierte die Ausführungsreihenfolge der Operationen bestimmt. Außerdem wird ein neuartiger Ansatz mit einem selbstadaptiven differentiellen Evolutionsalgorithmus vorgeschlagen, der durch eine Mutationsstrategie auf Basis der Nelder-Mead-Methode<sup>11</sup> weiter verbessert wird und die beiden zuvor genannten Ansätze übertrifft. Alle drei Ansätze zielen darauf ab, die Produktionszeit zu optimieren und gleichzeitig das zugehörige Scheduling mit der Zuordnung von Aufträgen zu Maschinenkonfigurationen abzuleiten. Dabei werden auch Zeitfenster für die Rekonfiguration der Maschinen berücksichtigt.

Auch [80] zielt auf die Minimierung der benötigten Zeit ab und stellt dazu ein Optimierungsmodell der linearen Programmierung für die Auslegung von zellularen rekonfigurierbaren Produktionssystemen (engl. Cellular Reconfigurable Manufacturing Systems - CRMS) vor. Dabei setzt sich die zu minimierende Zeit aus der Montage- und Demontagezeit der Hilfsmodule der RMTs, sowie aus der Zeit, die für den Transport der Teile zwischen den Zellen benötigt wird, zusammen. Hierzu sind sowohl die BOPs für die Fertigung unterschiedlicher Teile, die initiale Belegung der Zellen durch jeweils ein RMT, sowie die Transportzeiten zwischen den Zellen zentral gegeben.

Ein multikriterielles nichtlineares gemischt-ganzzahliges Programmierungsmodell wird in [81] vorgeschlagen. Dabei werden die Ziele Kosten, Rekonfigurationsaufwand und Qualitätsverschlechterung verwendet, wobei besonderes Augenmerk auf das Qualitätsziel und dessen Einfluss auf die beiden anderen Ziele gelegt wird. Zur Lösung des Modells verwenden die Autoren einen hybriden Ansatz, der einen *non-sorting genetic algorithm* (NSGA-II) für die Erkundung mit einer

---

<sup>10</sup> Der NSGA-II Algorithmus ist ein weitverbreiteter Vertreter aus der Klasse der evolutionary multi-objective optimization (EMO) Algorithmen. Diese Klasse zählt wiederum zu den a-posteriori-Methoden, welche darauf abzielen, alle pareto-optimalen Lösungen oder eine repräsentative Teilmenge der pareto-optimalen Lösungen zu finden.

<sup>11</sup> Auch bekannt als Downhill-Simplex-Verfahren.



*multi-objective particle swarm optimization* (MOPSO) für die Ausbeutung kombiniert<sup>12</sup>. Das zugrundeliegende RMS wird durch ein mehrstufiges Produktionssystem repräsentiert, in dem jeder Stufe eine Maschine zugeordnet ist. Ein Produktionsauftrag ist durch die Beschreibung der für ein einzelnes Produkt erforderlichen Operationen, sowie der Informationen über deren Vorrangigkeit, gegeben. Zudem wird in [81] aufgezeigt, dass viele Ansätze in der Literatur auf den Einsatz einzelner Heuristiken, insbesondere des NSGA-II, setzen und weniger oft mehrere Heuristiken im Rahmen eines einzelnen Ansatzes zum Einsatz kommen<sup>13</sup>. Ein Sekundäresultat von [81] ist dabei, dass für das formulierte Problem der eingesetzte NSGA-II bessere Ergebnisse liefert als die eingesetzte MOPSO.

Ein anarchistischer Produktionsansatz, indem ein verteiltes Planungs- und Steuerungssystem auf der Grundlage des freien Marktes vorgeschlagen wird, ist in [87] beschrieben. Insbesondere werden hier die Übergangs- und Anlaufphase mit einbezogen. Der anarchistische Ansatz basiert auf der Delegation der Entscheidungsfindung an die Elemente der untersten Ebene, d.h. Ressourcen, Jobs und Batch-Agenten verwenden einen verhandlungsbasierten Ansatz, um die Ressourcenallokation und die Reihenfolge der Operationen zu bestimmen. Eine detaillierte Betrachtung zur Generierung alternativer Systemkonfigurationen und zur Entscheidungsfindung über deren Aktivierung wurde bereits in Kapitel 2.4.1 vorgenommen, da in [87] ein Ansatz verfolgt wird, bei dem bereits für die Ermittlung des Rekonfigurationsbedarfs eine integrierte Durchführung der nachgelagerten Schritte des Rekonfigurationsmanagements erfolgt um rentable Rekonfigurationen identifizieren zu können.

Die simultane Berücksichtigung von Rüst- und Prozessplanrestriktionen wird im Zuge eines Ansatzes zur Generierung von integrierten Rüst- und Prozessplänen in [82] vorgestellt. Dabei werden die Gesamtkosten optimiert, wobei der Schwerpunkt auf den Rüstzeiten liegt. Der zweistufige Ansatz umfasst eine heuristische Generierung eines Rüstplans einschließlich einer Vorrichtungszuordnung, gefolgt von einem genetischen Algorithmus zur Generierung des besten Prozessplans auf der Grundlage dieses Plans und mit dem Ziel die genannten Gesamtkosten zu minimieren. Bei der Rüstplangenerierung werden zunächst sogenannte *Setup-Cluster* generiert, die jeweils eine Teilmenge der erforderlichen Operationen mit gleichen TADs unter Berücksichtigung ihrer Vorrangigkeit enthalten. Anschließend erfolgt die Vorrichtungszuordnung, bei der versucht wird, den verschiedenen Setup-Clustern mit gleicher TAD die gleiche Vorrichtung zuzuordnen. Im Zuge der Prozessplangenerierung werden den benötigten Operationen dann RMTs und deren Module, also Konfigurationsalternativen der Maschinenebene, zugeordnet.

---

<sup>12</sup> Weiter verbreitet sind die englischen Begriffe *Exploration* (für Erkundung) und *Exploitation* (für Ausbeutung).

<sup>13</sup> Ein Beispiel hierfür ist die Verwendung eines Genetischen Algorithmus und von Simulated Annealing innerhalb eines Ansatzes.

Ein mathematisches Modell für das Optimierungsproblem des Entwurfs von Maschinen mit Spindeln und Werkzeugrevolvern, die in mehreren Positionen arbeiten um unterschiedliche Produkttypen herzustellen, wird in [83] vorgestellt. Das Ziel ist die Optimierung der Gesamtkosten unter Berücksichtigung einer Vielzahl von Constraints. Diese Constraints sind z.B. die Vorrangigkeit der Operationen, die Zuordnung der Werkzeuge zu den Modulen und ihren Revolvern oder Spindeln und insbesondere die verschiedenen Orientierungsprobleme in Bezug auf die Werkstücke und die Werkzeuge. Das einkriterielle Optimierungsproblem wird mittels CPLEX gelöst und liefert den kostenoptimierten Maschinenentwurf als Ergebnis. Somit ist mit [83] ein Beispiel gegeben, dass sich auf die Maschinenebene fokussiert und die darüber hinausgehenden Aspekte der Systemebene nicht betrachtet.

Ein Ansatz, der eine Optimierung mit einer Materialfluss-Simulation zur Entwicklung und Auswahl optimierter Rekonfigurations-Szenarien kombiniert, wird in [138] präsentiert. Dazu wird ein Optimierungsmodell als deterministisches gemischt-ganzzahliges lineares Programm formuliert und mit CPLEX gelöst, um ein kostenoptimiertes Rekonfigurationsverfahren abzuleiten, das sowohl das Scheduling von Produktionsaufträgen als auch die notwendigen Rekonfigurationsmaßnahmen umfasst. Das Optimierungsmodell berücksichtigt dabei Produktions-, Lagerhaltungs-, Verzögerungs- und Rekonfigurationskosten. Um hingegen verschiedene Layouts für das Produktionssystem zu generieren und den Transportaufwand für jede Alternative zu berechnen, kommt der CRAFT-Algorithmus<sup>14</sup> zum Einsatz. Die ermittelten Rekonfigurationsverfahren und Layoutalternativen werden anschließend zu Rekonfigurations-Szenarien kombiniert, die jeweils in einer Materialfluss-Simulation ausgeführt und ausgewertet werden. Der Einsatz dieser Simulation erlaubt es, komplexe Zusammenhänge zu berücksichtigen, die in einer Optimierung nicht abgebildet werden können, wie z.B. die Verfügbarkeit von Wartungspersonal oder das Ausfallverhalten von Ressourcen.

Bei der Maschinenauswahl werden neben der reinen Auswahl der Maschinen häufig ebenso die entsprechenden Konfigurationsalternativen berücksichtigt und ausgewählt, so auch in [79–82, 87] und [89]. Damit besteht eine Abhängigkeit zur bereits beschriebenen Betrachtung der Maschinenebene. Weiterhin bildet die Auswahl der Maschinen die Grundlage für eine Allokation der Prozessschritte zu den Produktionsressourcen, wobei die Allokation und die Auswahl der Maschinen auch in einem Schritt durchgeführt werden können. So kann etwa im Zuge einer Optimierung die Allokation auf Basis der vorhandenen Maschinen bzw. deren Konfigurationsalternativen erfolgen, wie dies z. B. in [79–82, 89] geschieht. Aber auch Ansätze, die keine Optimierung beinhalten,

---

<sup>14</sup> Der Computerized Relative Allocation of Facilities Technique (CRAFT)-Algorithmus ist in diesem Kontext ein weit verbreiteter heuristischer Algorithmus.

können eine integrierte Auswahl und Allokation realisieren, wie z.B. [87] im Rahmen eines dezentralen, verhandlungsbasierten Ansatzes. Dabei kann eine Allokation wie in [80] auch mehrfach stattfinden, etwa wenn ein längerer Zeitraum betrachtet wird und die verschiedenen RMTs mehrfach rekonfiguriert werden können. Demgegenüber stehen Ansätze, die die Auswahl der Maschinen nicht berücksichtigen. Beispiele hierfür finden sich häufig unter den Ansätzen zur dynamischen Rekonfiguration, wie z.B. [86] aber auch bei Ansätzen mit reduziertem Betrachtungsraum, wie z.B. [83].

Sowohl für die Auswahl von Maschinen als auch für die Allokation ist der bereits für die Ermittlung des Rekonfigurationsbedarfs (s. Kapitel 2.4.1) beschriebene Abgleich von Produktionsanforderungen mit den Fähigkeiten des Produktionssystems relevant, wobei hier die Fähigkeiten der einzelnen Produktionsressourcen und von deren Konfigurationsalternativen relevant sind. Insgesamt ist also eine Wissensmodellierung und eine daraus resultierende Wissensrepräsentation erforderlich, die all diese Aspekte der Rekonfigurationsmanagement-Methodik ermöglichen kann und daher in Kapitel 2.3 ausführlich diskutiert wurde. In vielen Ansätzen werden hierzu die oben beschriebenen Informationen über die Konfigurationen bzw. Konfigurationsalternativen auf der Maschinenebene verwendet um diesen Abgleich basierend auf den benötigten Operationen durchzuführen. In selteneren Fällen, wie z.B. in [81], umfasst dies zudem den Abgleich der TADs von Konfigurationsalternativen und benötigten Operationen.

Für die Maschinenauswahl sowie auch die Anpassung des Produktionsprozesses im Sinne einer Allokation und der Ableitung einer Produktionssequenz wird in [23] die Idee eines generischen Ansatzes zur Selbstbeschreibung von CPPMs in einem produktgesteuerten Produktionssystem beschrieben. Es wird vorgeschlagen, zunächst ein Matching von Produktmerkmalen und Produktionstechnologien abzuleiten, dann eine Sequenz dieser Technologien und anschließend mögliche Produktionssequenzen, die dann das eingesetzte Fabrik-Equipment umfassen. Produktmerkmale können dabei etwa *Löcher* sein, die unter anderem durch ihre *Tiefe* oder *Oberflächenrauigkeit* beschrieben werden.

Für die Positionierung von ausgewählten Maschinen im Layout des Produktionssystems ist anzumerken, dass in der Literatur unterschiedlichste Layout-Betrachtungen vorgenommen werden. So gibt es zunächst einige Ansätze die kein komplexes Layout sondern eine mehrstufige Produktionslinie betrachten [79, 81]. Dort wird oft, wie in [79], jeder Stufe eine Maschine (bzw. eine Maschinenkonfiguration) zugewiesen, die dann eine stufenspezifische Operation durchführt. Dies entspricht dann keiner tatsächlichen Positionierung im Layout sondern vielmehr einer Allokation.

Desweiteren gibt es Ansätze wie [89], bei denen zwar keine vordefinierten Produktionsstufen vorgesehen sind, es aber eine festgelegte Anzahl an Maschinen im Layout gibt und lediglich deren Konfigurationsalternativen geändert werden können.

In [80] werden unterschiedliche Layoutstrukturen über eine Varianz in der Anzahl an Produktionszellen und darauf basierend unterschiedliche Layoutvarianten über die Betrachtung der möglichen Maschinen-Zellen-Zuordnungen, also der Positionierung von RMTs in den Produktionszellen, berücksichtigt. Dabei heben die Autoren von [80] den großen Einfluss der unterschiedlichen Layoutvarianten, basierend auf ihren Ergebnissen, hervor.

In [138] werden, wie oben beschrieben, mit Hilfe des CRAFT-Algorithmus verschiedene Layouts für das Produktionssystem generiert. Durch einen paarweisen Austausch von Komponenten und eine Bewertung des Transportaufwands wird eine Verbesserung des Layouts erreicht. Eine Skalierung des Produktionssystems wird ermöglicht, indem entweder eine Station, eine Linie oder ein ganzer Bereich hinzugefügt oder entfernt wird, wenn die Kapazität des Produktionssystems angepasst werden muss.

Die Anpassung des Produktionsprozesses kann im Rahmen der Generierung von Alternativen auf der Systemebene über das Ableiten von Allokationen, Produktionssequenzen oder Parametrierungen, oder einer Schnittmenge aus diesen Aspekten adressiert werden.

So wird die Produktionssequenz in [76] basierend auf dem Produktionsauftrag auf der einen und einer ontologiebasierten Wissensbasis auf der anderen Seite abgeleitet. Anschließend findet eine Software-Reallokation statt, bei der die in IEC 61449-Funktionsblöcken gekapselten Steuerungsalgorithmen auf die Maschinensteuerungen aufgespielt werden. Zudem verwendet [77] Symbolische KI (engl. Symbolic AI) als Schlüsseltechnologie und realisiert einen MAPE-K-ähnlichen Selbstadaptionsprozess, der auf einer eher abstrakten Ebene beschrieben wird und auf einer Skill-basierten Beschreibung von Fähigkeiten zur Sequenzplanung und Ressourcenallokation basiert. Um eine Produktionssequenz und eine Ressourcenallokation zu erhalten, wird in [88] ein auf Software-Containern basierender Ansatz für die dynamische Planung und Steuerung eines CPPS vorgeschlagen. Die Autoren verwenden einen zweistufigen Prozess, in dem zunächst eine optimierte Aufgabenplanung und -zuweisung an Maschinen oder Menschen festgelegt wird, um anschließend die Struktur der Steuerungssoftware mit Hilfe von Funktionsblöcken und Containern zu orchestrieren und den jeweiligen Geräten zuzuweisen.

Basierend auf einem Katalog an Fertigungsservices kann ein Ressourcen-Agent in [86] unter Berücksichtigung von Kontextregeln und einer maximalen Anzahl installierbarer Services einen Pool von zulässigen alternativen Konfigurationen ableiten. Dabei werden Rekonfigurationsmaßnahmen in Form einer Parametrierung der Produktionsressource oder des Austauschs von Services eingeplant, um eine Allokation und Parametrierung des Produktionsprozesses in Bezug auf eine vorgegebene Produktionssequenz zu finden.

Wie bereits beschrieben, erfolgt die Allokation häufig im Rahmen einer Optimierung, wie z. B. in [79–83] und [89], kann aber auch unabhängig von einer Optimierung durchgeführt werden, wie z.B. in [87]. Das Ableiten einer Produktionssequenz wird wiederum teilweise im Rahmen einer

Optimierung durchgeführt [89], teilweise aber auch vernachlässigt, da z.B. eine BOP gegeben ist und die Produktionssequenz vorgibt [79]. In [80] werden sogar mehrere BOPs vorgegeben um unterschiedliche Teile fertigen zu können. Es existieren jedoch auch Ansätze wie [81], bei denen Vorrangsgraphen vorgegeben werden und Produktionssequenzen im Rahmen der dadurch gegebenen Restriktionen abgeleitet werden. Ein Beispiel für die Integration der Vorrangigkeit von Operationen mittels eines gerichteten Graphen ist in [83] gegeben.

Eine intelligente Exploration des Lösungsraums bei der Generierung alternativer Konfigurationen kann im Wesentlichen über eine Zerlegung des Gesamtproblems in Teilprobleme, welche dann dezentral adressiert werden, sowie über eine mehr oder weniger intelligent ausgeprägte Reduktion des Lösungsraums adressiert werden. Es gibt jedoch auch Ansätze, die diesen Aspekt vernachlässigen und lediglich eine Systemkonfigurationsalternative oder ein definiertes Ensemble aus diesen ableiten.

So wird bei der Generierung von Konfigurationsalternativen (auf der Maschinenebene) in [86] ein Matching-Algorithmus mit dem Ziel der Reduktion des Pools an möglichen Rekonfigurationsalternativen von Fertigungsservices eingesetzt. Dieser analysiert, unter Berücksichtigung des Kontexts und der aktuellen Situation und ergänzt um maschinelles Lernen und semantisches Reasoning (mit Hilfe des JENA-Reasoners) jede Rekonfigurationsalternative, um nicht gangbare, nicht sinnvolle sowie ungünstige Lösungen zu identifizieren und verwerfen zu können. Im Detail wird geprüft, ob ein Satz von Attributen eines Services mit den technischen Rahmenbedingungen der Produktionsressource (z.B. der Durchmesser des neuen Bohrers muss kleiner sein als der maximal von der Maschine unterstützte Durchmesser) vereinbar ist. Dazu werden Services, Ressourcen und Prozesse semantisch im RDF/XML-Format beschrieben. Das Ergebnis ist eine Ansammlung von gangbaren Service-Rekonfigurationsalternativen, die anschließend bewertet und in eine Rangfolge eingeordnet werden können. Wie in Kapitel 2.1 beschrieben, zielt [86] demnach nicht auf eine globale Optimierung ab, sondern vielmehr auf ein globales Management auf Systemebene.

Es gibt aber auch Ansätze wie z.B. [78], die auf eine schnelle Identifizierung einer willkürlichen gangbaren Lösung abzielen und darauf verzichten eine optimale oder optimumnahe Lösung durch eine aufwändige Exploration des Lösungsraums zu finden. Hierzu schlagen Ansätze wie [124] sogenannte *best practices* als Lösungen vor, welche zuvor durch Literaturrecherchen und Experteninterviews mit Anlagenbetreibern identifiziert wurden.

Eine Reduktion des Lösungsraums wird oft über den Einsatz von Constraints im Rahmen einer Optimierung wie z.B. in [79–83] und [89] bewerkstelligt. So kann sichergestellt werden, dass nur gangbare Lösungen betrachtet werden oder auch gezielt Lösungen mit ungewollten Eigenschaften exkludiert werden. So wird etwa in [82] durch jeweils einen Constraint sichergestellt, dass die

gefundenen Lösungen alle benötigten Operationen umfassen, dass diese einer Konfigurationsalternative auf Maschinenebene zugeordnet sind und dass die Vorrangigkeit der Operationen eingehalten wird. Des Weiteren werden teilweise sogenannte *Lower Bounds* (untere Schranken) eingesetzt, diese werden oft, wie z.B. in [89] mit Hilfe einer Relaxation ermittelt und ermöglichen es Bereiche des Lösungsraums zu identifizieren, die nicht exploriert werden müssen<sup>15</sup>.

Ein Beispiel für die Zerlegung des Gesamtproblems in Teilprobleme, wird in [87] gegeben, indem Ressourcen, Jobs und Batch-Agenten einen verhandlungsbasierten Ansatz zur dezentralen Entscheidungsfindung verfolgen um die Ressourcenallokation und die Reihenfolge der Operationen zu bestimmen.

### 2.4.3 Bewertung und Auswahl von Konfigurationen

In diesem Teilkapitel werden Ansätze für die Bewertung und Auswahl von Konfigurationen beschreiben. In vielen Ansätzen sind die Bewertung und Auswahl eng mit der Generierung von alternativen Konfigurationen (Kapitel 2.4.2) verzahnt, insbesondere da dort oft eine Optimierung zum Einsatz kommt, welche bereits auf die Zielkriterien für die Bewertung und Auswahl abzielt. Demzufolge wurden Aspekte dieses Kapitels bereits in Kapitel 2.4.2 angeführt. In [87] wird gezeigt, dass sogar Ansätze existieren bei denen die Beschreibung der Ermittlung des Rekonfigurationsbedarfs (Kapitel 2.4.1) nicht sinnvoll erfolgen kann, ohne die Aspekte der Bewertung und Auswahl bereits anzuführen, sofern diese wie in [87] in die Ermittlung des Rekonfigurationsbedarfs integriert werden. Es lässt sich also festhalten, dass die Bewertung und Auswahl zwar wesentliche Bestandteile eines umfassenden Rekonfigurationsmanagements sind, die auch in vielen Ansätzen enthalten sind, sie sind jedoch nicht immer als explizite nachgestellte Phasen realisiert.

Auf der übergeordneten Ebene lassen sich entsprechend der in Kapitel 1.3 definierten Anforderungen die beiden Aufwandstypen Rekonfigurations- und Produktionsaufwand unterscheiden. Im Folgenden wird daher aufgeführt in welcher Form die beiden Aufwandstypen in der Literatur adressiert werden. Hierzu ist anzumerken, dass die Bewertung und Auswahl prinzipiell den Vergleich der Aufwände gegenüber dem resultierenden Nutzen einer Rekonfiguration berücksichtigen sollte, wie dies schon in Kapitel 1.2 angeführt wurde. Wobei der Nutzen bei vielen Ansätzen in der Literatur durch die Realisierung der Produktionsanforderungen bzw. der Produktionsaufträge vorgegeben ist und sich der Vergleich entsprechend auf die Betrachtung der Aufwände re-

---

<sup>15</sup> Lower Bounds sind insbesondere durch die Branch-and-Bound-Methode bekannt. Bei dieser wird der Lösungsraum in Teilbereiche unterteilt. Werden Lower Bounds für diese Teilbereiche berechnet und sind schlechter als die beste bisher gefundene Lösung, so kann dieser Teilbereich des Lösungsraums ignoriert werden.

duziert. Die Rekonfigurationsaufwände können zudem hardware- oder softwareabhängige Aufwände sein. Bevor jedoch explizite Kriterien zur Spezifikation dieser Aufwände genannt werden, sei darauf hingewiesen, dass in der Literatur neben den quantitativen auch qualitative Ansätze für diesen Aspekt verfolgt werden. So wird mitunter die Anzahl der Änderungen als Maß verwendet, um eine minimalinvasive Rekonfiguration zu erreichen, wovon man sich eine aufwandsarme Rekonfiguration verspricht. In diesem Zusammenhang wird z.B. in [78] die Anzahl der zu maskierenden Eingangsvariablen, welche in diesem Ansatz der Anzahl an Rekonfigurations-Eingriffen entspricht, minimiert, was tendenziell, aber nicht notwendigerweise den Rekonfigurationsaufwand minimiert.

In Bezug auf softwareabhängige Rekonfigurationsaufwände (auf der Maschinenebene) werden in [86] die Rekonfigurationskosten auf Basis von durchschnittlichen und nicht näher spezifizierten Kostenwerten für das Entfernen, Hinzufügen oder Verbessern eines Services berechnet. Dabei werden die aktuelle Konfiguration und die betrachtete alternative Konfiguration als Vektoren beschrieben, um die Anzahl der zu entfernenden, hinzuzufügenden und zu verbessernden Services und die resultierenden gesamten Rekonfigurationskosten zu ermitteln. Außerdem werden die Verfügbarkeit, die Ausführungszeit und der Durchsatz aller einzelnen Services einer betrachteten Konfigurationsalternative (Maschinenebene) abgeschätzt, um den implizierten Nutzen einer Rekonfiguration zu quantifizieren.

Der in [85] vorgeschlagene Ansatz beinhaltet ein mathematisches Modell, das darauf abzielt, das optimale Scheduling in Bezug auf den Rekonfigurationsaufwand (unter Verwendung eines vereinfachten Ersatzwertes), die Minimierung der Lieferverzögerung und die Minimierung des Arbeitsvolumens auf vereinfachte Weise zu finden. Ein *multi-objective self-organizing migratory algorithm* wird verwendet, um die Menge der pareto-optimalen Lösungen abzuleiten, aus denen ein Entscheidungsträger wählen kann. Das Ergebnis ist eine Reihenfolge für die Herstellung der benötigten Produkte, wie sie in den gegebenen Aufträgen gefordert werden.

Einige Ansätze in der Literatur verwenden Beispielwerte als Basis für die Bestimmung der Rekonfigurations- und/oder Produktionsaufwände. Ansätze wie [87] führen zu diesem Zweck sogar eine Pseudo-Währung ein. So wird der zeitliche Aufwand für die Rekonfiguration auf der Maschinenebene in [89] basierend auf einer Normalverteilung zwischen 75 und 150 bestimmt, um Beispielwerte zu erhalten. Auch die Bestimmung der Ausführungszeiten für Operationen der Konfigurationsalternativen der Maschinen geschieht in [89] nach demselben Prinzip. Beide Aufwandstypen werden im Zuge der Optimierung von [80] und [89], die bereits für die Generierung der Konfigurationsalternativen auf der Systemebene, wie sie in Kapitel 2.4.2 beschrieben wurde, berücksichtigt, um die Produktionszeit zu minimieren. Da es sich sowohl in [89], wie auch in [80], um Beispiele für eine einkriterielle Optimierung handelt, die neben den Produktionsaufwänden auch die Rekonfigurationsaufwände integriert und zu lediglich einer einzigen optimierten Lösung führt, ist kein zusätzlicher Schritt für eine weitere Bewertung und Auswahl notwendig und das

Optimierungsergebnis kann angewendet werden. Dagegen werden in [80] im Rahmen einer sogenannten Multi-Szenario-Analyse zusätzlich verschiedene Layoutvarianten betrachtet, so dass sich zunächst mehrere Lösungen ergeben, die jeweils anhand ihrer interzellularen Transportzeiten sowie der kombinierten Montage- und Demontagezeiten der RMTs bewertet werden und eine subsequente Auswahl erforderlich ist. Ähnliches gilt in [138], hier werden die Ergebnisse der unterschiedlichen Simulationsläufe in einem KPI-Dashboard präsentiert um darauf basierend eine manuelle Auswahl der besten Alternative zu ermöglichen. Dabei werden etwa die Anzahl der produzierten Teile und die Kosten pro Teil sowie die Overall Equipment Effectiveness (OEE)<sup>16</sup> angezeigt.

In [79] werden wie bereits in Kapitel 2.4.2 beschrieben die Produktionsaufwände als Produktionskosten in \$ berechnet, indem die Kosten aller RMT-Konfigurationen, die einer jeweiligen Produktionsstufe der mehrstufigen Produktionslinie zugeordnet wurden, aufsummiert werden. Neben der Minimierung dieser Produktionskosten werden die Rekonfigurierbarkeit, die Flexibilität und die Zuverlässigkeit maximiert, indem ein NSGA-II zur Generierung einer (großen) Menge an nicht-dominierten Lösungen verwendet wird. Zur Auswahl der besten Lösung wird dieser NSGA-II mit einer TOPSIS-Methode<sup>17</sup> kombiniert, die die Shannon-Entropie-Theorie (SET) zur Gewichtung und Einstufung der abgeleiteten nicht-dominierten Lösungen integriert. Die Grundidee von TOPSIS ist, dass die ausgewählte Alternative den kleinsten euklidischen Abstand zur positiven Ideallösung und den größten euklidischen Abstand zur negativen Ideallösung haben sollte. Die positive Ideallösung wird durch eine hypothetische Alternative mit den besten Werten für alle Kriterien aus der Menge aller gefundenen Lösungen dargestellt. Die negative Ideallösung entspricht einer hypothetischen Alternative mit den schlechtesten Werten für alle Kriterien aus der Menge aller gefundenen Lösungen. So werden im angeführten Beispiel acht optimierte nicht-dominierte Lösungen (Chromosomen) durch den NSGA-II generiert und anschließend die beste Lösung mit Hilfe der beschriebenen Methode ausgewählt.

Darüber hinaus werden in [81] die Ergebnisse einer Literaturrecherche vorgestellt, in der untersucht wurde, anhand welcher Kostenfaktoren die Kosten von RMS in der Literatur analysiert werden. Demnach werden in der Literatur sowohl einzelne Kostenfaktoren als auch Kosten, die sich aus mehreren Faktoren zusammensetzen, betrachtet, wobei die Betrachtung der Produktionskosten am weitesten verbreitet ist [81, 139–141]. Zusammengefasst diagnostiziert [81], dass die Analyse von RMS-Kosten meist auf Produktionskosten, Maschinennutzungskosten oder (Re-)Konfi-

---

<sup>16</sup> Im Deutschen auch als Gesamtanlageneffektivität bezeichnet, gibt die OEE die Verfügbarkeit, Leistung und Qualität eines Produktionssystems an.

<sup>17</sup> Technique for Order Preference by Similarity to Ideal Solution (TOPSIS).



gurationskosten beruht. Zudem werde oft die Zeit als Analysekriterium berücksichtigt, wohingegen die Qualität und insbesondere die Auswirkungen der Qualität auf die Kosten nicht betrachtet würden [81]. Daher werden in [81] wie in Kapitel 2.4.2 beschrieben die Kosten, der Rekonfigurationsaufwand und Qualitätsverschlechterung für die Bewertung im Zuge der Generierung alternativer Konfigurationen verwendet, wobei besonderes Augenmerk auf das Qualitätsziel und dessen Einfluss auf die beiden anderen Ziele gelegt wird. Dazu setzen sich die in [81] zu minimierenden Kosten aus den Faktoren Produktionskosten, Maschinennutzungskosten, Ausschusskosten, Nacharbeitskosten und Rekonfigurationskosten (auf der Systemebene) zusammen. Zudem werden der *Modularity Effort*, der die Zeit für die Rekonfiguration auf der Maschinenebene, d.h. für das Hinzufügen, Entfernen oder Modifizieren von Hilfsmodulen, berücksichtigt, sowie der *Quality Decay Index*, der das Verhältnis von fehlerhaften zu konformen Operationen ausdrückt, in Constraints überführt und als Zielkriterien bei der Optimierung berücksichtigt. Insgesamt werden nicht-dominierte Lösungen ermittelt wobei die subsequente Auswahl dem Betreiber überlassen wird.

Bezüglich der Prominenz von Bewertungskriterien (und demzufolge oft auch Optimierungskriterien) ergaben die Recherchetätigkeiten im Rahmen dieser Forschungsarbeit, ähnlich wie in [81], dass Zeit und (monetäre) Kosten mit Abstand am weitesten verbreitet sind und ansatzspezifisch aus den unterschiedlichsten Faktoren zusammengesetzt werden. Im Gegensatz zu [81] konnte jedoch auch die Fertigungsqualität als ein weit verbreitetes Kriterium identifiziert werden. Darüber hinaus wurde ein Trend hin zu der Betrachtung der Energie identifiziert. Dieser ist mitunter auf die gestiegene Aufmerksamkeit für den CO<sub>2</sub>-Fußabdruck - welcher teilweise direkt berücksichtigt wird - und auf den Klimawandel insgesamt zurückzuführen. Diese und weitere Erkenntnisse wurden auch im Rahmen der bereits zu Beginn von Kapitel 2 erwähnten systematischen Literaturanalyse zum Thema Rekonfigurationsmanagement erarbeitet und in [74] publiziert.

In [82] basiert die einkriterielle Optimierung im Rahmen des zweistufigen heuristischen Ansatzes, welche zugleich eine Bewertung und Auswahl für die unterschiedlichen Alternativen bewerkstelligt, wie in Kapitel 2.4.2 beschrieben, auf der Minimierung von Gesamtkosten, welche sich aus unterschiedlichen Kostenfaktoren zusammensetzen. Diese Kostenfaktoren werden in [82] unterteilt in die Bearbeitungskosten für die Durchführung der geforderten Operationen, die Toleranzkosten für die benötigte Präzision bei voneinander abhängigen Operationen, die Kosten für einen Wechsel des Setups und die Kosten für einen Modulwechsel (innerhalb des generierten Prozessplans). Ähnliches gilt für die einkriterielle Optimierung der Gesamtkosten in [83], wobei sich diese in [83] aus der Multiplikation der Kostenfaktoren mit der Anzahl der entsprechenden Betriebsmittel hinsichtlich einer Position, eines Werkzeugrevolvers, eines Bearbeitungsmoduls oder eines Spindelkopfes ergeben.

Zudem existieren auch Ansätze, bei denen der Nutzen nicht über die reine Realisierung der Produktionsaufträge gegeben ist. Diesbezüglich befasst sich etwa [142] ausführlich mit der Nutzenschätzung durch eine Flexibilitätsbewertung und der zugehörigen Literatur. Dort werden auch

verschiedene Ansätze zur Messung von Teilflexibilitäten behandelt. Ein hoher Nutzen kann dabei durch eine starke Verbreiterung des Flexibilitätskorridors eines Produktionssystems erreicht werden.

Die Auswahl von Konfigurationen ist eng mit der vorhergehenden Bewertung verzahnt und basiert im Wesentlichen auf den Ergebnissen eben jener Bewertung der Konfigurationsalternativen. Für die Auswahl können drei Kategorien, anhand dessen wie die Entscheidungsfindung unterstützt wird, unterteilt werden. (1) Es findet keine Auswahl statt, (2) es werden mehrere Alternativen vorgeschlagen, aber der Betreiber muss die Auswahl treffen oder (3) es werden mehrere Alternativen berücksichtigt und die beste Alternative wird dem Betreiber vorgeschlagen. Vertreter dieser drei Kategorien wurden bereits in Kapitel 2.1 im Zuge der Betrachtung von Anforderung 1.4 angeführt.

In [86] wird ein Ansatz zur verteilten Entscheidungsfindung vorgestellt, ob eine zuvor ermittelte beste Rekonfigurationsalternative (auf Maschinenebene) angewendet werden soll oder nicht. Dazu wird zum einen ein sogenannter *Nervositäts-Stabilitäts-Mechanismus* und zum anderen ein *Kollaboratives Interaktionsprotokoll* eingesetzt. Der *Nervositäts-Stabilitäts-Mechanismus* jedes Ressourcenagenten begrenzt dazu die Häufigkeit der Service-Rekonfigurationen, wobei diese Häufigkeit mit Hilfe des Q-Learning-Algorithmus [143] angepasst wird. Das *Kollaborative Interaktionsprotokoll* definiert die FIPA-ACL-konforme Agentenkommunikation und soll sicherstellen, dass auf Systemebene keine notwendige Servicefunktionalität verloren geht, wenn dieser Service durch die Rekonfiguration einer Ressource entfernt werden soll. Darüber hinaus wird eine nicht näher spezifizierte Simulation mehrerer Szenarien verwendet, um den Nutzen einer Rekonfigurationsalternative auf Systemebene zu überprüfen.

## Diskussion

Neben den bereits in Kapitel 2.1 beschriebenen Erkenntnissen aus der quantitativen Literaturliteraturauswertung zur Erfüllung der Anforderungen werden im Folgenden einige für die Konzeption dieser Forschungsarbeit relevante Erkenntnisse aus der qualitativen Betrachtung der Literatur hervorgehoben. Diese ergeben sich insbesondere aus Kapitel 2.4, werden aber teilweise auch durch gewonnene Einsichten aus den vorangegangenen Teilkapiteln sowie aus der bereits erwähnten systematischen Literaturliteraturanalyse [74] angereichert.

Wie bereits in Kapitel 2.3 in Bezug auf die zugrunde liegende Modellierung gezeigt wurde, existieren bereits zahlreiche Ansätze, um ein Matchmaking von Produktanforderungen mit den Fähigkeiten eines Produktionssystems zu ermöglichen. Dieses Matchmaking ist sowohl für die *Ermittlung von Rekonfigurationsbedarf* als auch für die *Generierung alternativer Konfigurationen* essenziell. Hierfür hat sich die Verwendung des PPR-Konzepts etabliert, auf dessen Basis jedoch ein geeignetes Konzept entwickelt werden muss, um den Anforderungen an die Konzeption ge-

recht zu werden. Grundsätzlich ist für die funktionale Modellierung zu entscheiden, ob ein *prozess-orientierter* oder eine *interface-orientierter* Ansatz verfolgt werden soll. Folgt man dem weitverbreiteten *prozess-orientierten* Ansatz, wird ein Produkt anhand der zu seiner Herstellung notwendigen Prozesse, also anhand einer BOP, beschrieben. Dies hat den Vorteil, dass für einen eingehenden Produktionsauftrag schnell und einfach geeignete Produktionsressourcen gefunden werden können [129]. Nachteilig hingegen ist die Problematik, dass die Produktdesigner unter Umständen keine Kenntnisse über die Produktionsprozesse haben und somit Aufwand für die Identifikation der geeigneten Prozesse entsteht [144]. Außerdem ist eine einheitliche (bzw. standardisierte) Prozessbeschreibung erforderlich. Der in [126] vorgeschlagene *interface-orientierte* Ansatz stellt hingegen das Produkt in den Vordergrund. Hierbei werden die Prozessoperatoren mit Hilfe von Merkmalsräumen der Eingangs- und Ausgangszustände der Produkte beschrieben. Dies hat den Vorteil, dass keine BOP benötigt wird und eine prozessunabhängige Beschreibung vorliegt. Nachteile ergeben sich daraus, dass das Problem der einheitlichen Beschreibung auf eine einheitliche Produktbeschreibung verlagert wird, die unter Umständen aktualisiert werden muss, wenn neue Produkteigenschaften auftreten, die zum Zeitpunkt der Modellierung noch nicht bekannt waren [144]. Ein potenzieller Nachteil ist, dass deutlich höhere Anforderungen an die Produktionsplanung gestellt werden, da erst vor Beginn der Produktion entschieden wird, wie das Produkt hergestellt werden soll. Dieser potenzielle Nachteil kann aber auch als Vorteil gesehen werden, da sich mehr Freiheitsgrade für eine Generierung von Produktionssequenzen ergeben. Diese kann wiederum durch die Intelligenz des CPPS und seiner CPPMs adressiert werden.

In Bezug auf die *Ermittlung des Rekonfigurationsbedarfs* können unterschiedliche Trigger für eine Untersuchung des Vorliegens von Rekonfigurationsbedarf berücksichtigt werden, die sich in *interne* und *externe* Trigger-Arten kategorisieren lassen. Während für die Betrachtung externer Trigger in der Literatur dominant der Eingang eines neuen Produktionsauftrages adressiert wird, wird für die Betrachtung interner Trigger häufig der Ausfall von zuvor gegebener Funktionalität gewählt. In dem zu entwickelnden Konzept soll dies bei der *Ermittlung des Rekonfigurationsbedarfs* berücksichtigt werden.

Insbesondere im Hinblick auf die *Generierung alternativer Konfigurationen* zeigt sich, dass die Literatur von Ansätzen mit einer *klassischen*, maschinenbaulichen Sichtweise geprägt ist. Dem Konzept der RMS und damit verbunden den RMT kommt dabei besondere Bedeutung zu. Da diese, wie in Kapitel 1.2 beschrieben, Wurzeln der CPPS darstellen, fließt diese Erkenntnis, insbesondere im Hinblick auf die Modularität, in das zu entwickelnde Konzept ein. Des Weiteren werden verschiedene Formen der Optimierung zur gezielten Generierung von Konfigurationsalternativen eingesetzt, wobei hier ein Trend hin zur Verwendung von Heuristiken wie z.B. genetischen Algorithmen zu erkennen ist. In Bezug auf die Betrachtung unterschiedlicher Layoutvarianten wurde deren großer Einfluss in [80] hervorgehoben.

Die *Bewertung und Auswahl von Konfigurationen* soll entsprechend der Anforderungen multikriteriell erfolgen, wobei die Kriterien Zeit und (monetäre) Kosten in der Literatur als weit verbreitet identifiziert wurden. Zudem konnte ein Trend hin zur Betrachtung der Energie festgestellt werden. Dies gilt ebenso im Hinblick auf die Optimierungskriterien, wenn eine multikriterielle Optimierung zum Einsatz kommt. Erfolgt dabei eine Approximation der Pareto-Front bzw. wird eine Menge von nicht-dominierten Lösungen generiert, so wird zudem eine Methode zur Auswahl einer dieser Lösungen benötigt. Des Weiteren gibt es Ansätze, die hinsichtlich des Nutzens einer Rekonfiguration auf eine (generelle) Verbreiterung des Flexibilitätskorridors abzielen. Nachteilig ist hierbei jedoch, dass der tatsächliche Nutzen eines breiteren Flexibilitätskorridors schwer zu quantifizieren ist, da zukünftige Produktionsaufträge und die damit verbundenen benötigten Funktionalitäten noch nicht bekannt sind.

Ergänzend zu den Erkenntnissen aus Kapitel 2.1 bezüglich der geringen „Nutzung der Potenziale von cyber-physischen Produktionssystemen“ durch die untersuchten Ansätze ist folgendes festzuhalten. Die meisten Ansätze nutzen weder den Vernetzungsaspekt, noch Modelle, sondern häufig eine Tabelle, die globale, meist exemplarische, Informationen beinhaltet. In dieser Hinsicht wird beispielsweise in [89] eingeräumt, dass eine Betrachtung der realen Welt an dieser Stelle sinnvoll wäre. Daher würde der Ansatz stark von der Verwendung von Modellen und einem CPS-basierten Ansatz profitieren. Darüber hinaus wird in vielen Publikationen, wie z.B. [145], selbst wenn ein CPS-basierter Ansatz verfolgt und Modelle verwendet werden, eher ein globales Modell mit allen Informationen über die Ressourcen verwendet, als dezentrale modulspezifische Modelle. Zudem gibt es einige Ansätze, wie z.B. [89], die zwar als CPS-basiert bezeichnet werden, aber den Vernetzungs- bzw. Kommunikationsaspekt nicht nutzen oder dies zumindest nicht erwähnen. Zusammenfassend lässt sich zudem sagen, dass [89] und [146] zwei von vielen Beispielen darstellen, in denen die Autoren eher Ansätze beschreiben, die *für* CPPS genutzt werden können, als Ansätze die auf CPPS *basieren*. Insbesondere bei den *klassischen* Ansätzen beschränken sich die Modelle, sofern überhaupt Modelle verwendet werden, häufig auf die funktionalen Fähigkeiten der Produktionsressourcen und die Beschreibung der notwendigen Operationen zur Transformation eines Produkts, nicht aber auf Verhaltensaspekte, nicht-funktionale Eigenschaften oder Materialfluss- bzw. Transportaspekte. Daher tendieren viele *klassische* Ansätze dazu, mathematische Modelle abzuleiten, oft auf elegante Weise, aber eher für sehr spezifisch definierte Probleme, und zu versuchen, diese entweder exakt (z.B. [147, 148]) oder mit Hilfe von Heuristiken (z.B. [149, 150]) zu lösen. Simulationsbasierte Ansätze für CPPS (wie z.B. [151, 152]) sind dagegen derzeit weniger verbreitet. Das Verhalten des realen Produktionssystems im Sinne seiner Steuerungslogik wird generell von den meisten Ansätzen vernachlässigt. Dies gilt auch für Ansätze, die auf dem CPS-Konzept basieren, die diesen Aspekt jedoch häufiger abdecken als die eher *klassischen* Ansätze. Eine der wenigen Ausnahmen ist z.B. [153], wobei [153] das Verhalten berücksichtigt und in Form von Automaten für jede Maschine und jedes Transportsystem modelliert. Dabei können

simulationsbasierte Ansätze auch eingesetzt werden um das Verhalten des realen Produktionssystems im Sinne seiner Steuerungslogik zu berücksichtigen. Dies führt zu einer besseren Abbildung der Realität, da das Steuerungsverhalten deterministisch statt stochastisch berücksichtigt werden kann. Damit werden auch die Ergebnisse des Rekonfigurationsmanagements im Sinne der Realitätsnähe robuster.

*Insgesamt kann auch festgehalten werden, dass sich zahlreiche Forschungsaktivitäten mit Teilaspekten des betrachteten Forschungsgegenstandes befassen, was dessen Relevanz unterstreicht.*

### 3 Konzept für das selbstorganisierte Rekonfigurationsmanagement cyber-physischer Produktionssysteme

Um die in Kapitel 1.3 formulierte Forschungsfrage beantworten zu können wird in diesem Kapitel das entwickelte Konzept, als das wesentliche Artefakt dieser Forschungsarbeit vorgestellt, das hierzu die abgeleiteten Anforderungen erfüllen soll. Die Entwicklung des Artefakts fand entsprechend des Design Science Researchs in mehreren Iterationen statt, aus Gründen der Übersichtlichkeit und Nachvollziehbarkeit werden jedoch lediglich die finalen Resultate dargestellt. Da im Rahmen des Rigor Zyklus der Wissensbasis neue wissenschaftliche Erkenntnisse hinzugefügt werden sollen, sind viele Inhalte dieses Kapitels, insbesondere in [30–32, 75] und [154], bereits im Rahmen dieser Forschungsarbeit publiziert worden.

#### 3.1 Übersicht über die Bestandteile des Konzepts

Um cyber-physische Produktionssysteme um die Fähigkeit eines selbstorganisierten Rekonfigurationsmanagements bereichern zu können, müssen wie in Kapitel 1.3 beschrieben, verschiedene Resultate zur Beantwortung der dort formulierten drei Teilfragen erarbeitet werden. Im Sinne des Design Science Researchs bilden diese drei zu entwickelnde Teilartefakte (vgl. Kapitel 1.5). Die konkreten Teilartefakte sind in Abbildung 3.1 dargestellt. Das erste und wichtigste Teilartefakt bildet die zu entwickelnde **Methodik** für das selbstorganisierte Rekonfigurationsmanagement, diese wird in Kapitel 3.2 detailliert beschrieben. Um jedoch ein selbstorganisiertes Rekonfigurationsmanagement zu ermöglichen, ist ein geeigneter Ansatz zur Wissensmodellierung und -verwaltung von entscheidender Bedeutung. Daher stellt die zu erarbeitende **Modellierung** das zweite Teilartefakt dar und wird in Kapitel 3.3 erläutert. Das dritte Teilartefakt bildet die **CPPS-Architektur** die geeignet definiert werden muss, um die Wissensmodellierung und das Wissensmanagement sowie die Methodik zu integrieren. Die CPPS-Architektur wird in Kapitel 3.4 behandelt und umfasst dabei auch das zugrunde liegende Produktionssystem und dessen Steuerung.

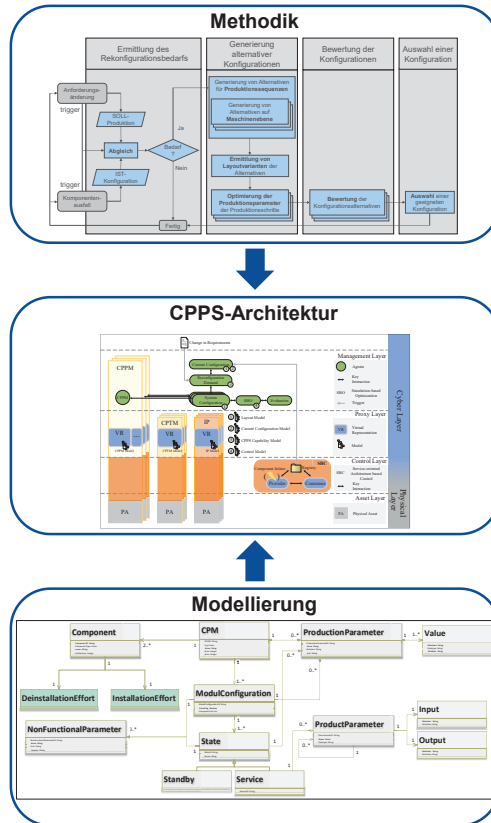


Abbildung 3.1: Bestandteile des Konzepts alias Teilartefakte

### 3.2 Methodik für das selbstorganisierte Rekonfigurationsmanagement

Um eine geeignete Abbildung der Schritte des Rekonfigurationsmanagements im CPPS (s. Kapitel 1) zu ermöglichen muss zunächst eine geeignete Methodik entworfen werden. Dementsprechend wird die Ausprägung der in Kapitel 1 definierten und durch die abgeleiteten Anforderungen A1.1-A1.4 eingeforderten Schritte des Rekonfigurationsmanagements: *Ermittlung des Rekonfigurationsbedarfs*, *Generierung alternativer Konfigurationen*, *Bewertung der Konfigurationen* und *Auswahl einer Konfiguration* in den nachfolgenden Teilkapiteln beschrieben. Eine Übersicht über die gesamte resultierende Methodik für das selbstorganisierte Rekonfigurationsmanagement ist in Abbildung 3.2 dargestellt.

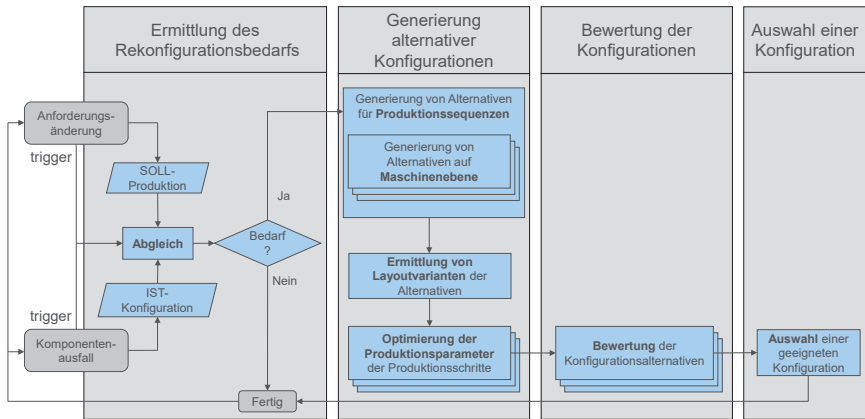


Abbildung 3.2: Rekonfigurationsmanagement-Methodik

### 3.2.1 Ermittlung des Rekonfigurationsbedarfs

Um ein selbstorganisiertes Rekonfigurationsmanagement zu ermöglichen, muss es zunächst möglich sein, einen bestehenden Rekonfigurationsbedarf im laufenden Betrieb identifizieren zu können. Der linke Teil von Abbildung 3.2 zeigt eine vereinfachte Beschreibung des Verfahrens das die Ermittlung des Rekonfigurationsbedarfs ermöglicht.

Dabei gibt es zwei verschiedene Auslöser (Trigger), die zu einer Überprüfung auf einen potenziell vorhandenen Rekonfigurationsbedarf führen können. Zum einen können sich Anforderungsänderungen durch neu erteilte Produktionsaufträge ergeben. Zum anderen können Komponentenausfälle innerhalb des Produktionssystems eine Überprüfung auslösen.

Geänderte Anforderungen wirken sich auf die SOLL-Produktion aus, während sich Komponentenausfälle auf die IST-Konfiguration auswirken. Wann immer einer der beiden genannten Trigger auftritt, wird die potenziell geänderte SOLL-Produktion mit der potenziell geänderten IST-Konfiguration verglichen. Es findet also ein Abgleich der Produktionsanforderungen mit den Fähigkeiten des CPPS statt.

Die SOLL-Produktion besteht aus der Beschreibung der jeweils geforderten Produkte, die hierfür eingesetzte funktionale Modellierung basiert auf der interface-orientierten Formalisierten Prozessbeschreibung (FPB) [126]. Diese wurde insbesondere deshalb gewählt, weil sie, wie bereits in Kapitel 2.4 erwähnt, zusätzliche Freiheitsgrade für eine intelligente Generierung von Produktionssequenzen durch das CPPS und seine CPPMs bietet. Außerdem enthält die SOLL-Produktion die Gewichtung der festgelegten Bewertungskriterien, die hier aufgrund der Erkenntnisse aus Kapitel 2.4 zu Zeit, Kosten und Energie gewählt wurden.



Die IST-Konfiguration enthält eine Beschreibung aller aktuellen Produktionsressourcen (CPPMs) des CPPS. Dazu gehören deren Fähigkeiten zur Durchführung von Produktionsprozessen, einschließlich der einstellbaren Parameterbereiche und Produktionskapazitäten. Zusätzlich umfasst die IST-Konfiguration die Positionierung der aktuellen CPPMs innerhalb des Layouts.

Ergibt der Vergleich, dass die SOLL-Produktion nicht mit der IST-Konfiguration durchgeführt werden kann, erfolgt die Generierung alternativer Konfigurationen, die in Kapitel 3.2.2 beschrieben wird, andernfalls wird auf das Eintreten des nächsten Triggers gewartet.

Um den Vergleich einer SOLL-Produktion mit der IST-Konfiguration des CPPS zu ermöglichen, wird wie beschrieben die interface-orientierte FPB als konzeptionelle Grundlage für die Modellierung verwendet. Die Fähigkeiten einer Produktionsressource (jeweils Teil des entsprechenden CPPM), die diese, wie in Kapitel 3.3 detaillierter ausgeführt wird, in Form von aufrufbaren Services zur Verfügung stellt, werden durch einen Prozessoperator beschrieben, der zwei Listen von Merkmalen und deren Werte definiert. Die erste Liste beschreibt die Anforderungen an den Produktionseingang, z.B. die Breite und Länge, die ein Eingangsprodukt erfüllen muss, um von der Produktionsressource bearbeitet werden zu können. Die zweite Liste beschreibt die möglichen Ausgangsprodukte, die eine Produktionsressource produzieren kann. Es werden also die möglichen Transformationen, die eine Produktionsressource an dem jeweiligen Eingangsprodukt durchführen kann, beschrieben.

Abbildung 3.3 zeigt, wie die FPB verwendet wird, um das CPPS-Fähigkeitsmodell (engl. *CPPS Capability Model*) zu realisieren, das die IST-Konfiguration des CPPS beschreibt. In dem abgebildeten Beispiel bieten die Ressourcen des CPPS die drei Prozessoperatoren Stanzen, Bohren und Fräsen (engl. *stamp*, *drill* und *mill*) mit den entsprechenden Beschreibungen der Merkmalsräume der Eingangs- und Ausgangszustände der Produkte an. Eingangsmerkmale, die mit *preValues* versehen sind, werden verwendet, um anzuzeigen, dass Werte von einem Vorgänger übernommen werden. Bei Ausgangsmerkmalen zeigen *preValues* an, dass der Prozessoperator das entsprechende Merkmal nicht verändert.

Auf der Grundlage dieses Beschreibungsschemas wird das Fähigkeitsmodell erstellt, indem der Ausgang eines Prozessoperators mit dem Eingang eines anderen Prozessoperators verglichen wird, was in einer Baumstruktur mit Sequenzen von Produktionsprozessen resultiert. Die Wurzeln (roots) dieser Baumstruktur sind die Ausgänge der Menge der verfügbaren Prozessoperatoren. Sie definieren alle möglichen Ausgänge der aktuellen Konfiguration des CPPS auf generische Art und Weise. Jeder mögliche Ausgang ist mit dem entsprechenden Prozessoperator verbunden, wobei der jeweilige Eingang (orange) Beschränkungen in Bezug auf Prozessoperatoren die angehängt werden können enthält. Diese Eingänge werden dann mit den Ausgängen aller nächstmög-



*Generierung alternativer Konfigurationen* im Rahmen dieser Forschungsarbeit entwickelte *intelligente Exploration des Lösungsraums* zudem in entsprechend verallgemeinerter Form in [154] publiziert. Dort wird der Begriff der *Filter* eingesetzt, der im Folgenden aufgegriffen wird, um die Abstraktionsfähigkeit des Ansatzes auch in dieser Ausarbeitung aufzuzeigen<sup>18</sup>.

Demnach wird das übergeordnete Problem, also das finden alternativer Systemkonfigurationen, in sequentiell voneinander abhängige Teilprobleme zerlegt wird, die über die nachfolgend beschriebenen Teilschritte *Generierung von Alternativen für Produktionssequenzen*, *Ermittlung von Layoutvarianten der Alternativen* und *Optimierung der Produktionsparameter der Produktionsschritte* adressiert werden. Jeder Teilschritt resultiert in einer Anzahl möglicher Lösungen, die zunächst zu einer horizontalen Unterteilung führen, die verschiedene, eigenständige Teil-Lösungsräume darstellen. Über die Durchführung der drei Teilschritte, werden schließlich verschiedene Lösungen für das gesamte, übergeordnete Problem gefunden. Jeder einzelne Lösungsrepräsentant, also jede betrachtete alternative Systemkonfiguration, prägt sich über sich über die Teilschritte hinweg selbständig aus, indem er die jeweiligen Lösungsunterräume exploriert. Der Detaillierungsgrad bzw. die Vollständigkeit der Lösungen nimmt dabei mit jedem durchlaufenen Teilschritt zu. In jedem Teilschritt können dabei unterschiedliche Mechanismen (Filter) zur gezielten Reduktion des Lösungsraums eingesetzt werden, es werden also drei Filterstufen durchlaufen. Daraus ergibt sich eine Baum-Struktur, welche das übergeordnete Problem in verschiedene Teilprobleme (vertikal) und verschiedene Lösungsteilräume (horizontal) zu den jeweiligen Teilproblemen unterteilt.

Für die Mechanismen zur Filterung des Lösungsraums kommen bis zu drei unterschiedliche Typen der Filterung, sogenannte *Filterlevel*, in jeder Filterstufe zum Einsatz. Die drei *Filterlevel* basieren dabei jeweils auf der Anwendung einer oder mehreren Filterregeln und dienen unterschiedlichen Zwecken, die nachfolgenden beschrieben werden:

- Das 1. *Filterlevel* kann genutzt werden, um unerwünschte Lösungen, so genannte "No-Gos", noch vor der Exploration des jeweiligen Teil-Lösungsraumes auszuschließen. Diese No-Gos sind nicht-funktionaler Natur, d.h. sie beruhen auf Qualitätskriterien und ihre Verwendung bietet die Möglichkeit, den Teil-Lösungsraum innerhalb einer Filterstufe frühzeitig zu begrenzen. Die Anwendung des 1. Filterlevels ist optional.

---

<sup>18</sup> Eine weitergehende Übersicht über die im Rahmen dieser Forschungsarbeit entwickelten universellen Methodik für die intelligente Exploration des Lösungsraums, deren spezifische Ausprägung für das Rekonfigurationsmanagement sowie über die Filterlevel ist im Anhang aufgeführt.

- Das 2. *Filterlevel* dient dazu, die funktionalen Anforderungen an die Lösung des jeweiligen (Teil-)Problems zu formulieren. Der Einsatz des 2. Filterlevels ist somit eine obligatorische Maßnahme mit dem Ziel, den Teil-Lösungsraum so zu reduzieren, dass im weiteren Verlauf nur noch sinnvolle Lösungen berücksichtigt werden.
- Das 3. *Filterlevel* wird zur Formulierung weicher Kriterien für die jeweilige Problemlösung genutzt. Sie ermöglicht es, die gefundenen Lösungen in eine Rangordnung zu überführen und damit eine intelligente Auswahl treffen zu können, die den (Teil-)Lösungsraum weiter reduziert. Das 3. Filterlevel ist optional.

Hierbei stellt die *Generierung von Alternativen für Produktionssequenzen* den ersten Teilschritt der Generierung von alternativen Konfigurationen dar und ist beispielhaft in Abbildung 3.4 dargestellt. Zur Vereinfachung wird die FPB um eine generische Eingangs- und Ausgangsproduktbeschreibung ergänzt, die durch unterschiedlich gestaltete rote Formen repräsentiert wird. Basierend auf dem Produktionsauftrag in Form einer Eingangs- und Ausgangsproduktbeschreibung (siehe Production Order 1 in Abbildung 3.4) werden Produktionssequenzen ermittelt, die in der Lage sind, die Transformation des gegebenen Eingangsprodukts in das gewünschte Ausgangsprodukt durchzuführen. Dies entspricht einer Filterung des Lösungsraums hinsichtlich der funktionalen Machbarkeit.

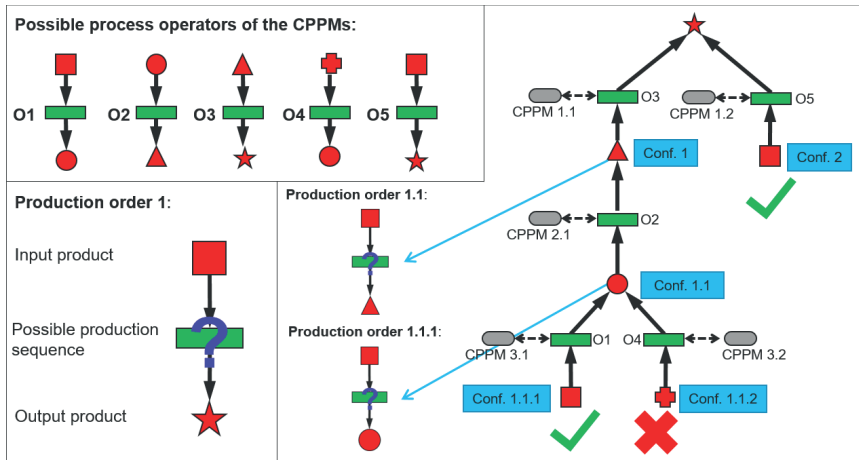


Abbildung 3.4: Verfahren zur Generierung von Alternativen für Produktionssequenzen (Beispiel) [31]

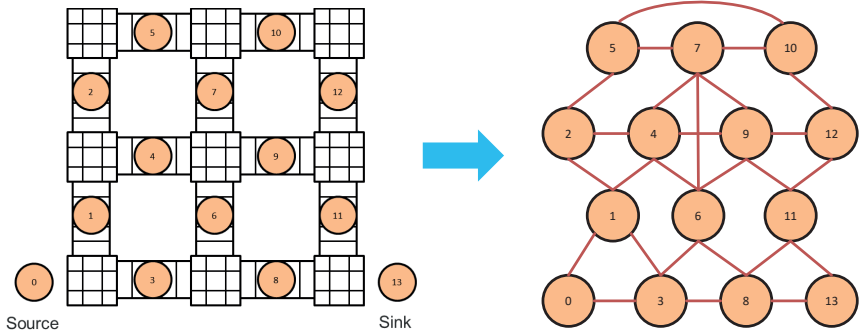
Zunächst wird allen CPPMs die Beschreibung des gewünschten Ausgangsprodukts zur Verfügung gestellt. Die CPPMs führen die *Generierung von Alternativen auf Maschinenebene* durch, wobei sie ermitteln, ob sie in ihrer aktuellen oder einer alternativen Konfiguration einen Prozessoperator (O) anbieten können, der ein beliebiges Eingangsprodukt in das gewünschte Ausgangsprodukt

verarbeiten kann (siehe O3 und O5 in Abbildung 3.4). Dies geschieht mit Hilfe eines Abgleichs des gewünschten Ausgangsprodukts mit den jeweiligen Prozessoperatoren, welche wie bereits in Kapitel 3.2.1 erläutert, über ihre Merkmale beschrieben sind. Anschließend wird für jeden der gefundenen Prozessoperatoren der CPPMs eine neue Systemkonfiguration (z.B. Conf 1 und Conf. 2) erstellt und der Prozessoperator sowie seine zugehörige CPPM-Konfiguration mit dem Ausgangsprodukt verbunden. Danach wird wieder nach dem gleichen Schema, also beginnend mit der Bereitstellung der Beschreibung des nun benötigten Ausgangsproduktes an die CPPMs (siehe Production Order 1.1 in Abbildung 3.4), versucht, Prozessoperatoren mit dem Eingangsprodukt jedes zuvor verbundenen Prozessoperators zu verbinden. Dies geschieht so lange, bis das Eingangsprodukt eines hinzugefügten Prozessoperators mit dem angeforderten Eingangsprodukt übereinstimmt, eine Schleife auftritt oder kein geeigneter Prozessoperator mehr gefunden werden kann.

Hierbei ist es auch möglich die oben erwähnte Filterung des Lösungsraums über die Exklusion von No-Gos durchzuführen. So können beispielsweise einzelne CPPMs oder auch komplette alternative Systemkonfigurationen, die einen bestimmten Grenzwert für den Energieverbrauch im Standby überschreiten, von der weiteren Betrachtung ausgeschlossen werden.

So werden die unterschiedlichen Systemkonfigurationen durch den dezentralen, rückwärtsgewandten Ansatz erstellt und prägen sich fortan selbstorganisiert weiter aus.

Nach der Ermittlung der alternativen Produktionssequenzen müssen jeweils die Möglichkeiten die CPPMs im Layout des Produktionssystems zu positionieren durch die *Ermittlung von Layoutvarianten der Alternativen* untersucht werden. Immer wenn eine neue, alternative, Layoutvariante gefunden wurde, wird diese durch eine neu erzeugte Systemkonfiguration repräsentiert, welche sich wiederum selbstorganisiert weiter ausprägt. Hierzu wird das Layout des CPPS als Graph abgebildet, in dem die möglichen Maschinenstandorte nummeriert und die möglichen Transportverbindungen dargestellt sind. Dies ist in Abbildung 3.5 beispielhaft für das Layout des modularen Produktionssystems dargestellt, das auch in der Evaluierung verwendet wird.



**Abbildung 3.5: Abbildung der Layout-Struktur des modularen Produktionssystems als Graph**

In dieser zweiten Filterstufe kann ein genetischer Algorithmus eingesetzt werden, um für jede zuvor gefundene alternative Systemkonfiguration, d.h. innerhalb jedes Teil-Lösungsraumes, eine einstellbare Anzahl vielversprechender Layoutvarianten (unter Verwendung der Filterlevel 2 und 3) zu finden. Unterschiedliche Ausprägungen des Filterlevels 3, z.B. mit dem Ziel eines optimierten Transportaufwandes, eines reduzierten Rekonfigurationsaufwandes oder einer Kompromisslösung aus beidem, können eingesetzt werden, um das Finden der Lösungen zu beeinflussen. Alternativ kann ein Brute-Force-Ansatz verwendet werden, um alle möglichen Layoutvarianten für die zuvor gefundenen alternativen Systemkonfigurationen zu generieren. Darüber hinaus besteht die Möglichkeit, die generierten Layoutvarianten vorsortieren zu lassen, so dass die vielversprechendsten Layoutvarianten im weiteren Verlauf zuerst ausgeprägt werden. Dazu erfolgt für die mit der Brute-Force-Methode (potenziell) erzeugten Layoutvarianten eine Bewertung hinsichtlich des Transportweges, um eine Vorsortierung zu ermöglichen. Eine detaillierte Beschreibung der Realisierung der Ermittlung der Layoutvarianten erfolgt in Kapitel 4.1.

Basierend auf den Layout-Informationen der aktuellen Konfiguration, den Rekonfigurationsaufwänden der CPPMs, die auf Basis der Installations- und Deinstallationsaufwände aus den Modellen der CPPMs (siehe Kapitel 3.3) für jede Komponente eines CPPMs ermittelt werden, und der ermittelten Layoutvariante werden die Rekonfigurationsaufwände für jede alternative Systemkonfiguration berechnet. Diese Rekonfigurationsaufwände  $a_{z,reconfiguration}$  beschreiben die Aufwände der Rekonfigurationsmaßnahmen um von der aktuellen in die jeweils betrachtete alternative Systemkonfiguration zu überführen und werden für jedes der vorgegebenen Kriterien  $z$  (Zeit, Kosten und Energie) berechnet, so dass ein späterer Vergleich ermöglicht wird. Optional können anschließend Lösungen verworfen werden, wenn sie einen einstellbaren Grenzwert hinsichtlich der Rekonfigurationsaufwände überschreiten.

Im Ergebnis sind nun alle unterschiedlichen Layoutvarianten durch eine entsprechende alternative Systemkonfiguration repräsentiert.

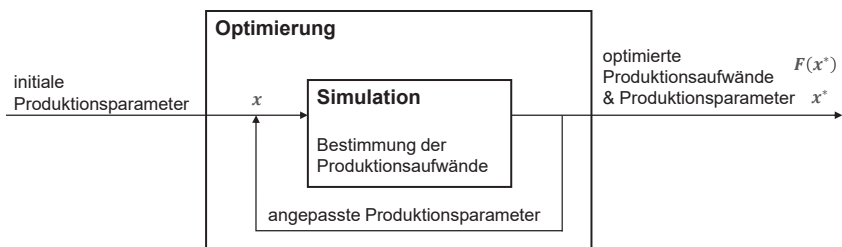
An dieser Stelle sei erwähnt, dass dieses Konzept prinzipiell auch die Möglichkeit bietet, verschiedene Layout-Strukturen in die Betrachtung mit einzubeziehen. So könnten diese miteinander verglichen und die optimale Layout-Struktur für die gegebenen Anforderungen gefunden werden.

Im letzten Teilschritt der Generierung alternativer Konfigurationen wird die *Optimierung der Produktionsparameter der Produktionsschritte* für jede alternative Systemkonfiguration durchgeführt.

Ziel ist es, den Produktionsaufwand  $a_{production}$  unter Berücksichtigung der gewichteten Kriterien (Zeit, Kosten und Energie) für jede alternative Systemkonfiguration zu minimieren. Hierzu muss berücksichtigt werden, dass CPPSs eine Steuerungslogik verwenden, die entsprechend Kapitel 1.2 auch nach einer Rekonfiguration noch greifen soll. Aufgrund der Komplexität dieser CPPS-Steuerungslogik kann die Zielfunktion der Optimierung jedoch nicht in einer geschlossenen Form ausgedrückt werden.

Der VDI stellt hierzu in der *VDI-Richtlinie 3633 Simulation von Logistik-, Materialfluss- und Produktionssystemen (Blatt 12 – Simulation und Optimierung)* klar: „Lässt sich der Zusammenhang zwischen Zielgröße und Einflussgrößen nicht explizit in einer Formel ausdrücken, ist man darauf angewiesen, die Werte der Zielfunktion in Abhängigkeit von ihren Einflussgrößen mittels eines Simulationsmodells zu ermitteln“ [155].

Daher kommt eine simulationsbasierte (multikriterielle) Optimierung zum Einsatz. Diese wird durch einen Ansatz der Kategorie D entsprechend der genannten VDI-Richtlinie [155] umgesetzt, bei dem eine hierarchische Verknüpfung durch die Integration der Simulation innerhalb der Optimierung erreicht wird (s. Abbildung 3.6).

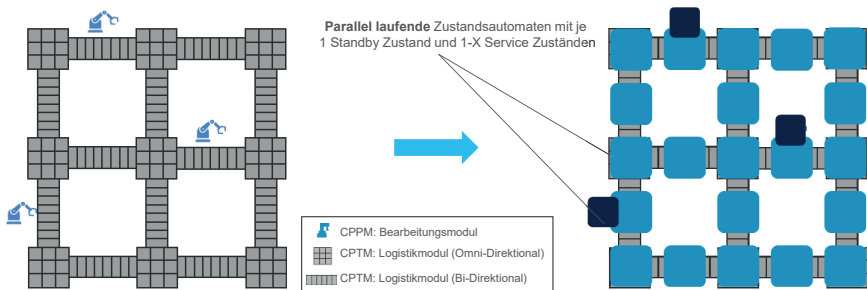


**Abbildung 3.6: Hierarchische Verknüpfung von Simulation und Optimierung nach Kategorie D - VDI 3633 (Blatt 12) [155]**

Dabei nimmt die Optimierung die führende Rolle ein, indem sie den Satz an Produktionsparametern  $x$ , welcher der Simulation übergeben wird in einem iterativen Verfahren variiert. Nach einem ersten Simulationslauf mit initial gewählten Startwerten (Satz an Produktionsparametern), erge-

ben sich erste Werte der Zielgröße im Sinne der resultierenden Produktionsaufwände. Diese werden nach jedem Simulationslauf durch die Optimierung ausgewertet und dabei solange angepasst bis ein definiertes Abbruchkriterium erfüllt wurde. Nach der finalen Iteration umfasst das Ergebnis der simulationsbasierten (multikriteriellen) Optimierung neben den optimierten Produktionsaufwänden  $F(x^*)$  auch die zugehörigen optimierten Produktionsparameter  $x^*$ .

Hierbei muss zunächst das Simulationsmodell erstellt werden, dabei erfolgt die Modellierung des Produktionssystems durch parallel laufende Zustandsautomaten als diskrete ereignisorientierte Simulation (s. Abbildung 3.7). Der Aufbau eines entsprechenden Simulationsmodells geschieht im Rahmen der entwickelten Methodik für jede vorliegende alternative Systemkonfiguration spezifisch, wenn auch nach gleichem Schema. Die Produktions- und Transportmodule der CPPMs und cyber-physischen Transportmodule (CPTMs) werden hierzu jeweils über einen Standby und 1-X Service Zustände modelliert, wobei die Anzahl der Service Zustände von der Anzahl der jeweils angebotenen Services eines cyber-physischen Moduls (CPMs) abhängt. Für den Aufbau der Simulationsmodelle werden die Informationen der jeweiligen Konfigurationsalternativen aus den CPM-Modellen (s. Kapitel 3.3) der eingesetzten CPMs verwendet.



**Abbildung 3.7: Modellierung des modularen Produktionssystems durch Zustandsautomaten**

Zudem erfolgt die Abbildung der Steuerungslogik des realen CPPS innerhalb des Simulationsmodells (weitere Details hierzu s. Kapitel 3.3 und Kapitel 4.1).

Um die resultierenden Produktionsaufwände (Zeit, Kosten, Energie) ermitteln zu können und deren Optimierung über eine Variation des Satzes an Produktionsparametern zu erlauben wird nachfolgend die Bildung der Ausgangssignale  $F(x)$  der Simulation erläutert, welche dann den resultierenden Produktionsaufwänden entsprechen.



Dabei berechnen sich die Service- und Standby-Produktionsaufwände  $f_{n,z,service}$  beziehungsweise  $f_{n,z,standby}$  für ein Kriterium  $z$  wie folgt:

$$f_{n,z,service} = \sum_{k=1}^p g_n^z(c_{n,z,service}, k) \quad (1)$$

$$f_{n,z,standby} = \sum_{k=1}^s g_n^z(c_{n,z,standby}, k) \quad (2)$$

Wobei  $g_n^z(\cdot)$  die Mapping-Funktion für alle Einflussfaktoren auf das Kriterium darstellt,  $c_{n,z,service}$  bzw.  $c_{n,z,standby}$  aus dem CPM-Modell der jeweiligen Modulkonfiguration stammt und über alle auftretenden Service-Zyklen  $p$  bzw. Standby-Zyklen  $s$  für das betrachtete Modul  $n$  aufsummiert wird.

Der gesamte Produktionsaufwand  $f_{n,z,tot}$  eines jeweiligen Moduls  $n$  in Bezug auf die Kriterien  $z$  kann dann wie folgt bestimmt werden:

$$f_{n,z,tot} = f_{n,z,standby} + f_{n,z,service} \quad (3)$$

Die Ausgangssignale des Simulationsmodells, die die resultierenden Produktionsaufwände in Bezug auf das jeweilige Kriterium  $z$  darstellen, können dementsprechend wie folgt berechnet werden:

$$f_z = \sum_{n=1}^{n_{max}} f_{n,z,tot} \quad (4)$$

wobei über alle  $n$  Module aufsummiert wird. Zu beachten ist dabei allerdings, dass die Berechnung des Zeitkriteriums  $f_{time}$  eine Ausnahme bildet und hierbei schlicht die folgende Formel verwendet wird:

$$f_{time} = f_{1,time,tot} \quad (5)$$

da die Summe aller Standby- und Servicezykluszeiten eines jeden Moduls (z.B. von Modul 1) bereits der Zeit entspricht, die für die Fertigstellung des gesamten Produktionsauftrags benötigt wird.

Entsprechend dieser Beschreibung können die Simulationsmodelle der jeweiligen Systemkonfigurationsalternativen dynamisch und vollautomatisiert aufgebaut werden.

Für die Realisierung der Optimierung können unterschiedliche Verfahren eingesetzt werden. So schlägt [155] etwa die Verwendung von Meta-Heuristiken wie den genetischen Algorithmen als Suchalgorithmen vor, da diese robust sind und keine Kenntnis über die topologischen Eigenschaften der Zielfunktion benötigen. Im Rahmen dieser Forschungsarbeit wurden insbesondere drei Verfahren implementiert aus denen ausgewählt werden kann. So ist es möglich, neben der in [32] detailliert beschriebenen Methode zur Lösung eines einkriteriellen Ersatzproblems auch ein Ver-

fahren zur multikriteriellen simulierten Abkühlung (engl. Simulated Annealing) oder einen multikriteriellen Genetischen Algorithmus einzusetzen. Eine detaillierte Beschreibung ist dementsprechend in Kapitel 4 gegeben.

In dieser Filterstufe (bzw. diesem Teilschritt) wird das 2. Filterlevel dadurch umgesetzt, dass nur tatsächlich einstellbare Parameterwerte erlaubt werden, welche wie beschrieben aus den Informationen der jeweiligen Konfigurationsalternativen aus den CPM-Modellen (s. Kapitel 3.3) der eingesetzten CPMs stammen. Das Filterlevel 3 wird durch die Optimierungsziele und die manuell einstellbaren maximalen Iterationszahlen abgebildet. Die simulationsbasierte Optimierung setzt also beide Filterlevel gleichermaßen um.

Das Ergebnis dieses Teilschrittes ist ein Satz von optimierten Produktionsparametern  $x^*$  für jede alternative Systemkonfiguration sowie das jeweilige Optimierungsergebnis  $F(x^*)$  selbst, das später für einen Vergleich und die potenzielle Anwendung der jeweiligen Systemkonfiguration verwendet werden kann.

Somit sind durch die *Generierung alternativer Konfigurationen* alle in Frage kommenden alternativen Systemkonfigurationen des CPPS ermittelt worden und stehen zur Bewertung und zum Vergleich bereit. Die intelligente Exploration des Lösungsraums ist somit abgeschlossen, wobei jede resultierende alternative Systemkonfiguration, die ihre zugehörigen Layoutvariante sowie den optimierten Parametersatz beinhaltet, einer separaten endgültigen Lösung des übergeordneten Problems entspricht.

Die Methodik weist an dieser Stelle mehrere Vorteile auf: Die Problemzerlegung im Sinne des Divide-and-Conquer-Prinzips ermöglicht eine verteilte und parallelisierte Lösung der Teilprobleme mit reduzierter Komplexität. Die verschiedenen Filterstufen, die im Zuge der beschriebenen Teilschritte der *Generierung alternativer Konfigurationen* durchlaufen werden, begrenzen gezielt das Wachstum des Lösungsraumes und damit den Zeitbedarf und den Rechenaufwand. Darüber hinaus ermöglicht das 3. Filterlevel die adaptive Begrenzung des Lösungsraumes, z.B. in Abhängigkeit von der verfügbaren Rechenleistung.

Filter sollten jedoch mit Bedacht eingesetzt werden, denn obwohl die Definition restriktiverer Filterregeln zu einer schnelleren Reduktion des Lösungsraums und damit zu einer schnelleren Exploration des Lösungsraums führt, kann die frühzeitige Exklusion von Lösungen, die nur auf einer Teilsicht auf das übergeordnete Problem beruht, gute Gesamtlösungen ausschließen und damit die Qualität des Explorationsergebnisses verschlechtern. Im Allgemeinen sollte daher versucht werden, so spät wie möglich zu filtern, um sicherzustellen, dass wichtige Teile des Lösungsraums nicht vorzeitig ausgeschlossen werden, und so früh wie nötig, um den Rechenaufwand zu reduzieren.

### 3.2.3 Bewertung der Konfigurationen

Grundlage für die Bewertung der Konfigurationen sind die Optimierungsergebnisse, die die Gewichtung der Kriterien berücksichtigen, für alle zuvor ermittelten Konfigurationen. Außerdem wird der Aufwand für die Durchführung der Rekonfigurationsmaßnahmen berücksichtigt. Insgesamt wird für die Bewertung der Systemkonfigurationen eine Nutzwertanalyse durchgeführt. Dabei wird für jede Systemkonfiguration ein Nutzwert  $v$  basierend auf der Ermittlung eines Aufwands werts und der Kriteriengewichtung ermittelt.

Als erstes werden für jedes Bewertungskriterium  $z$  die Rekonfigurationsaufwände  $a_{z,reconfiguration}$  und die Produktionsaufwände  $a_{z,production}$  (bzw.  $f_z(x^*)$  mit dem durch die Optimierung bestimmten  $x^*$ ) einer Systemkonfiguration zum kriterienspezifischen Gesamtaufwand  $a_{z,tot}$  aufsummiert:

$$a_{z,tot} = a_{z,reconfiguration} + a_{z,production} \quad (6)$$

Anschließend werden für jedes Kriterium  $z$  der maximale Aufwands wert  $a_{z,tot,max}$  und der minimale Aufwands wert  $a_{z,tot,min}$  unter allen betrachteten Systemkonfigurationen ermittelt. Darauf aufbauend erfolgt die Zuweisung eines normierten Aufwands werts  $r_z$  für jeden kriterienspezifischen Gesamtaufwand  $a_{z,tot}$  zwischen dem Maximalaufwand  $r_z = 0$  und dem Minimalaufwand  $r_z = 1$  nach der folgenden Formel:

$$r_z = \frac{a_{z,tot} - a_{z,tot,max}}{a_{z,tot,min} - a_{z,tot,max}} \quad (7)$$

Jeder dieser normierten Aufwands werte  $r_z$  einer Systemkonfiguration wird dann mit der jeweiligen Kriteriengewichtung  $w_z$  gewichtet und mit den Ergebnissen der anderen Kriterien aufsummiert. Demnach wird der Nutzwert  $v$  für jede Systemkonfiguration wie folgt berechnet:

$$v = \sum_z w_z * r_z \quad (8)$$

### 3.2.4 Auswahl einer Konfiguration

Um die am besten geeignete Konfiguration zu ermitteln, werden die Systemkonfigurationen anhand ihres Nutzwertes  $v$  verglichen, wobei der höchste Wert von  $v$  die am besten geeignete Systemkonfiguration repräsentiert.

Das Ergebnis dieser Methodik ist die neue Konfiguration für das CPPS.

Die daraus resultierende neue Systemkonfiguration umfasst Informationen auf Maschinen- und Systemebene, die die jeweilige Konfiguration aller CPPMs und deren Positionierung innerhalb des CPPS-Layouts sowie die optimierten Produktionsparameter beinhalten. Folglich können diese

Produktionsparameter für die Produktion mit der neuen Konfiguration und basierend auf der entsprechenden Produktionssequenz (im Sinne einer parametrisierten BOP) angewendet werden.

Darüber hinaus ist es auch möglich, mehr als nur eine Konfiguration (beispielsweise die *Top 3*) als Ergebnis zu liefern und als Entscheidungsgrundlage anzubieten, falls dies gewünscht ist.

### 3.3 Modellierung für das selbstorganisierte Rekonfigurationsmanagement

Die Informationsmodellierung umfasst im Wesentlichen die beiden Informationsmodelle für die CPMs und die Produktionsaufträge. Das vorgeschlagene Informationsmodell für CPMs, also entweder CPPMs oder CPTMs, ist in Abbildung 3.8 vereinfacht dargestellt und stellt den Kern des Modellierungsansatzes dar. In den nachfolgenden Paragraphen werden die Klassen beschrieben, die für die Informationsmodellierung verwendet werden.

Ein *CPM* kann mehrere *Modulkonfigurationen* (engl. *Module Configurations*) haben (dies widerspiegelt sich in der Vielzahl der in Kapitel 3.4 in Abbildung 3.10 veranschaulichten virtuellen Repräsentationen) und hat Kenntnis darüber, welche Konfiguration aktiv ist und welche Komponenten für die jeweilige Konfiguration verwendet werden.

Für jedes *CPM* wird der Pool, der alle Hardware- und Software-Komponenten (engl. *Components*) für alle seine möglichen *Modulkonfigurationen* umfasst, mit einer 2..\* Multiplizität zugeordnet. Der jeweilige *Installations-* und *Deinstallationsaufwand* (engl. *Installation* und *Deinstallation Effort*) für jede *Komponente* in Bezug auf die gewählten nicht-funktionalen Kriterien wird zugeordnet. Aus Gründen der Übersichtlichkeit ist die Aufwandsbeschreibung hier vereinfacht dargestellt, da die kriterienbasierte Aufwandsbeschreibung noch weitergehende Aspekte beinhaltet. Weitere Details zu den Aufwänden werden in der Realisierung (Kapitel 4) und im Anhang gegeben.

Um das Verhalten eines CPM zu beschreiben, wird das Konzept der Zustandsautomaten wie in Kapitel 3.2 beschreiben eingesetzt, weshalb auch hier jede *Modulkonfiguration* einen *Standby-Zustand* (engl. *Standby State*) und einen oder mehrere *Service-Zustände* (engl. *Service State*) hat. Diese Informationen können genutzt werden, um ein Simulationsmodell der jeweiligen alternativen CPPS-Konfigurationen zu erstellen und anschließend die simulationsbasierte multikriterielle Optimierung als Teil des selbstorganisierten Rekonfigurationsmanagements durchzuführen. Um dies zu ermöglichen, können den *Modulkonfigurationen*, sowie den *Zuständen*, *nicht-funktionale Parameter* (engl. *Non-functional Parameters*) hinzugefügt werden. Die jeweiligen Formeln drücken aus, wie die nicht-funktionalen Parameter, z.B. der Energieverbrauch eines Service, ermittelt werden. Dabei können die Formeln auch aus mehreren *Produktionsparametern* (engl. *Production Parameters*) bestehen. Die *nicht-funktionalen Parameter* dienen als Zwischengrößen, aus denen schließlich der kriterienspezifische Aufwand ermittelt wird.

Darüber hinaus erlaubt das Informationsmodell das Hinzufügen von *Produktionsparametern* auf CPM-, *Modulkonfigurations-* oder *Zustands-Ebene*. Die 1..\*-Multiplizität der Assoziation mit

dem *Wert-Element* (engl. *Value*) ermöglicht es, auch mehrere Wertebereiche, sowie die zugehörigen individuellen Schrittgrößen, der *Produktionsparameter* festzulegen.

In Bezug auf die Modellierung der *Services* wurde, wie oben erwähnt, die interface-orientierte FPB basierend auf der VDI/VDE-Richtlinie 3682 [127] gewählt. Um einen Abgleich der Produktanforderungen mit den Fähigkeiten des CPPS, genauer gesagt mit denen seiner CPPMs, zu ermöglichen, werden die Fähigkeiten der Module durch Prozessoperatoren beschrieben. Jeder *Service*, den ein CPPM anbieten kann, ist dabei basierend auf dem entsprechenden Prozessoperator (VDI/VDE-Richtlinie 3682), der die möglichen Transformationen beschreibt, den eine Produktionsressource an einem entsprechenden Produkt durchführen kann, modelliert. Diese Transformationen werden auf der Grundlage einer Liste von Merkmalen für den Eingangszustand und einer Liste für den Ausgangszustand eines Produkts definiert. Um dies zu modellieren, werden sowohl die *Produktparameter* als auch die *Eingang-* und *Ausgang-Elemente* (engl. *Input* und *Output*) verwendet. Darüber hinaus ermöglicht der vorgeschlagene Modellierungsansatz die Verfeinerung von Merkmalen mit Untermerkmalen durch die reflexive Assoziation der *Produktparameter* (engl. *Product Parameters*) zu sich selbst.

Im Rahmen des selbstorganisierten Rekonfigurationsmanagements werden diese Informationen zur *Ermittlung des Rekonfigurationsbedarfs* sowie zur *Generierung alternativer Konfigurationen* genutzt (d. h. zur Bestimmung möglicher Produktionssequenzen auf der Grundlage aktueller und alternativer Modulkonfigurationen der CPPMs).

Da die CPTMs keine Transformation an den Produkten durchführen, decken ihre Modelle diesen Aspekt nicht ab. Um diese Tatsache abbilden zu können sind die *Produktparameter* über eine 0..\*-Multiplizität mit dem *Service-Element* verbunden.

Das Informationsmodell der Produktionsaufträge ist in Abbildung 3.9 dargestellt. Um das zuvor beschriebene zu ermöglichen, wird ein *Produktionsauftrag* (engl. *Production Order*) durch das zu produzierende *Produkt* (engl. *Product*) spezifiziert. Das *Produkt* wird analog zu Abbildung 3.8 über den dort dargestellten Teil mit den *Produktparametern* und den *Eingang-* und *Ausgang-Elementen* beschrieben. Der einzige Unterschied besteht darin, dass die *Eingang-* und *Ausgang-Elemente* auf einen konkreten Wert gesetzt werden. Außerdem wird dem *Produktionsauftrag* die *Gewichtung* (engl. *Weighting*) der verschiedenen (nicht-funktionalen) *Kriterien* (engl. *Criterion*) zugewiesen.

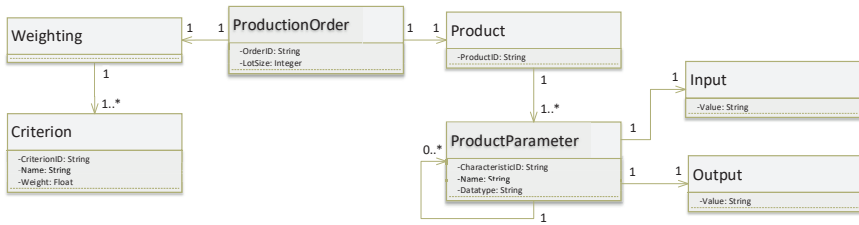


Abbildung 3.9: Informationsmodell der Produktionsaufträge [75]

Das *IP-Modell*, also das Modell eines Intelligenten Produkts (IP), wird in der CPPS-Architektur in Abbildung 3.10 als *IP-Model* bezeichnet und enthält eine BOP, die Teil des Ergebnisses des selbstorganisierten Rekonfigurationsmanagements ist und daher die entsprechenden Services und deren optimierte Produktionsparameter enthält. Dies steht im Kontrast zu anderen Ansätzen, bei denen der Produktionsauftrag die BOP enthält, und das IP-Modell aus dem Produktionsauftrag abgeleitet wird.

Des Weiteren werden die ersten drei CPPS-Modelle der CPPS-Architektur (s. Abbildung 3.10 in Kapitel 3.4) die dort in englischer Sprache als *Layout Model* (1), *Current Configuration Model* (2) und *CPPS Capability Model* (3) für die Ermittlung des Rekonfigurationsbedarfs verwendet, weshalb deren Verwendung bereits oben diskutiert wurde.

Zusammengefasst umfasst das *Layout Model* die Struktur des Produktions-Layouts und die mögliche Positionierung von CPPMs darin in Form eines Graphen. Das *Current Configuration Model* repräsentiert die Informationen der aktuell eingesetzten CPPS-Konfiguration und wird auf Basis der Discovery-Server-Informationen der SOA aktualisiert. Das *CPPS Capability Model* basiert auf den Informationen des *Current Configuration Models* und nutzt darüber hinaus die Informationen der CPPM-Modelle (engl. *CPPM Models*), um die Fähigkeiten des CPPS abzuleiten. Es enthält daher alle möglichen Produktionssequenzen, die von der aktuell eingesetzten CPPS-Konfiguration durchgeführt werden können (vgl. Kapitel 3.2.1), und wird aktualisiert, sobald Änderungen am *Current Configuration Model* auftreten. Das letzte verbleibende Modell ist das *Control Model* (4), welches das Steuerungskonzept des realen CPPS, also die SOA-basierte Steuerung (vgl. Kapitel 3.4) abbildet. Auf die Verwendung dieser vier Modelle und ihre konkrete Realisierung wird in Kapitel 3.4 und Kapitel 4 detaillierter eingegangen.

Wie aus der Diskussion zur Wissensmodellierung in Kapitel 2.3 hervorgeht, definiert das vorgeschlagene CPM-Informationsmodell die Vorlage für die eigentlichen Modelle der Systemkomponenten, insbesondere für die Produktionsmodule. Im Gegensatz zu den meisten Ansätzen in der Literatur umfasst der vorgestellte Modellierungsansatz auch nicht-funktionale Aspekte und ermöglicht somit die Anwendung von Methoden, die nicht-funktionale Kriterien einbeziehen und insbesondere das selbstorganisierte Rekonfigurationsmanagement.

Der vorgestellte Ansatz zur Informationsmodellierung kann darüber hinaus als ein erster Schritt zur Weiterentwicklung in Richtung einer Standardisierung genutzt werden, erhebt aber in seinem derzeitigen Zustand nicht den Anspruch, die notwendigen Standardisierungsaspekte umfassend abzudecken.

### **3.4 CPPS-Architektur für das selbstorganisierte Rekonfigurationsmanagement**

Die vorgestellte CPPS-Architektur sowie der Ansatz zur Wissensmodellierung und -verwaltung stehen nicht im Widerspruch zu bestehenden Ansätzen. Allerdings müssen die spezifischen Anforderungen, die sich bei der Realisierung von CPPS ergeben, die darüber hinaus in der Lage sein sollen, ein selbstorganisiertes Rekonfigurationsmanagement durchzuführen, berücksichtigt werden. Um dies zu erreichen, wurden mehrere Elemente aus bestehenden Ansätzen (siehe Kapitel 2.2) kombiniert. Die Ergebnisse werden im Folgenden beschrieben, wobei einige der Besonderheiten hervorgehoben werden.

Abbildung 3.10 visualisiert die, bereits in [75] publizierte, vorgeschlagene CPPS-Architektur und ihre Bestandteile für das selbstorganisierte Rekonfigurationsmanagement. Gemäß des Design Science Researchs wurde diese Architektur in einem iterativen Prozess im Sinne des Design Zyklus entwickelt, wobei im Sinne des Rigor Zyklus Erkenntnisse aus dem Stand der Wissenschaft und Technik eingeflossen sind. Daher stellt diese CPPS-Architektur eine Weiterentwicklung der in [30] präsentierten CPPS-Struktur und der Verortung der Intelligenz innerhalb des CPPS dar. Sie integriert den Wissensmodellierungs- und -verwaltungsansatz sowie die Funktionalität zur Realisierung der zur Umsetzung der Methodik notwendigen Intelligenz und erweitert damit CPPS um selbstorganisiertes Rekonfigurationsmanagement. Die integrierten Elemente bestehender Ansätze, die in Kapitel 2.2 beschrieben wurden, und die damit verbundenen Vorteile werden am Ende dieses Kapitels beschrieben.



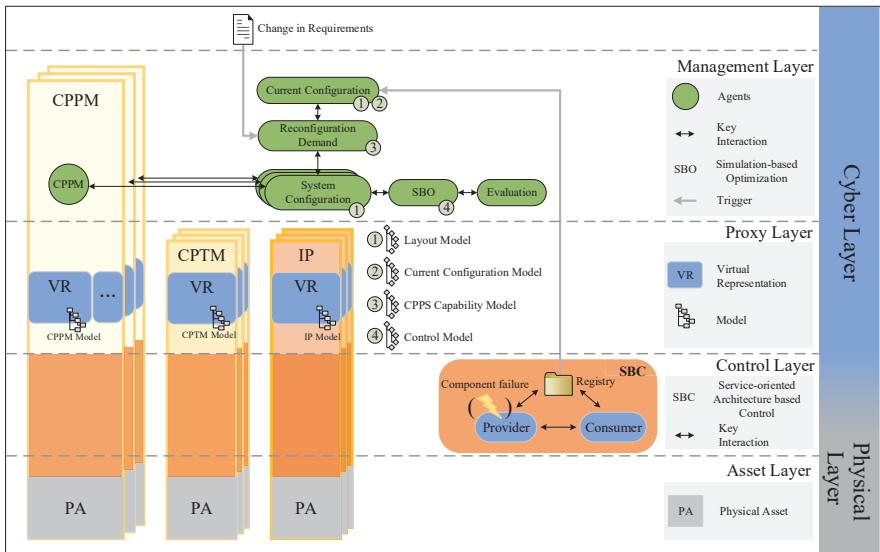


Abbildung 3.10: CPPS-Architektur für das selbstorganisierte Rekonfigurationsmanagement [75]

Die Architektur ist grundlegend in zwei Ebenen unterteilt: Diese grobe Unterteilung erfolgt in das *Physical-Layer* und das *Cyber-Layer*, wobei die erste Ebene die physischen Assets sowie deren Steuerungssoftware umfasst. Die zweite Ebene fügt den Cyber-Aspekt, insbesondere durch die Modelle und die Intelligenz des CPPS, hinzu. Die feingranulare Unterteilung erfolgt in das *Asset-Layer*, *Control-Layer*, *Proxy-Layer* und das *Management-Layer*. Diese Layer werden in Anlehnung an die Literatur und insbesondere an die RAMI 4.0 Spezifikation DIN SPEC 91345 [2] wie folgt beschrieben:

Das **Asset-Layer** repräsentiert die physische Welt und damit die Gesamtheit der real existierenden, so genannten *Physical Assets*. Dabei handelt es sich um physische Elemente wie Produktionsmodule, Transportmodule oder Produkte, die die physischen Anteile der *CPPMs*, *CPTMs* oder *IPs* darstellen.

Das **Control-Layer** erfüllt zwei Aufgaben. Zum einen dient es als Übergangsschicht von der physischen zur Informationswelt. Zum anderen ermöglicht es den Zugriff auf Informationen über ein einheitliches Datenformat und dient der Bereitstellung von Services, wie z.B. der Ausführung bestimmter Produktionsprozesse, mittels einer SOA. Diese SOA wird eingesetzt, um die Steuerung des CPPS zu realisieren, die notwendig ist, um die Produktionssequenzen für die verschiedenen Produktionsaufträge durchzuführen. Daher wird diese Steuerung als SOA-basierte Steuerung (engl. *SOA based Control*, kurz *SBC*) bezeichnet. In diesem Zusammenhang fungieren die CPPMs und CPTMs als *Anbieter* (engl. *Provider*) und die IPs als *Konsumenten* (engl. *Consumer*)

von Services. Das *Register* (engl. Registry) wird durch einen zentralen Discovery-Server realisiert, bei dem sich Serviceanbieter dynamisch an- und abmelden können. In Bezug auf RAMI 4.0 abstrahiert und verschmilzt das *Control-Layer* die beiden RAMI-Layer *Integration Layer* und *Communication Layer*.

Das ***Proxy-Layer*** enthält sowohl die *Virtual Representations* (VRs) der *Physical Assets* als auch die Sammlung weiterer Modelle. Das weitere für das MAS notwendige Wissen auf der Systemebene über das Layout (1), die aktuelle Konfiguration (2), die Fähigkeiten (3) und die Steuerung (4) des CPPS wird in den jeweiligen Modellen vorgehalten, die sich im *Proxy-Layer* befinden und von ihren zugehörigen Agenten (im *Management-Layer*) verwaltet werden können. Dieses Layer umfasst unter anderem die folgenden Aufgaben, von denen die meisten denen ähneln, die für das RAMI 4.0 *Information Layer* vorgesehen sind:

- Bereitstellung von formal beschriebenen Modellen.
- Persistieren der in diesen Modellen enthaltenen Daten.
- Sicherstellung der Datenintegrität.

Das ***Management-Layer*** umfasst die nötige Intelligenz, um die gewünschte Funktionalität für das Rekonfigurationsmanagement anzubieten. Digitale Repräsentanten aller Produktionsmodule sowie auch Repräsentanten von abstrakten Assets (ohne physisches Pendant) wie den Systemkonfigurationen interagieren auf selbstorganisierte Weise, um die möglichen neuen Systemkonfigurationen zu ermitteln und diese auszuprägen. Dieses Layer umfasst dementsprechend die Umsetzung der Methodik für das selbstorganisierte Rekonfigurationsmanagement (s. Kapitel 3.2) mit Hilfe eines MAS. Das *Management-Layer* ähnelt dem *Functional Layer* von RAMI 4.0, sowie dem *Application Layer*, welches, wie bereits in Kapitel 2.2 erwähnt wurde, in [97] beschrieben wird.

Die Softwareagenten, die innerhalb des *Management-Layers* eingesetzt werden, und ihre Interaktion mit dem Wissen, das durch das *Proxy-Layer* bereitgestellt wird, werden im Folgenden kurz vorgestellt:

- **Current Configuration Agent:** Verwaltet das *Current Configuration Model* (2) und das *Layout Model* (1) des CPPS. Außerdem stellt er dem *Reconfiguration Demand Agent* die aktuelle IST-Konfiguration zur Verfügung.
- **Reconfiguration Demand Agent:** Führt den Schritt *Ermittlung des Rekonfigurationsbedarfs* durch. Dazu reagiert er auf auftretende Trigger (d. h. Anforderungsänderungen oder Komponentenausfälle) mittels eines Vergleichs der SOLL-Produktion mit der aktuell eingesetzten IST-Konfiguration. Zu diesem Zweck erstellt und managt er das *CPPS Capability Model* (3) basierend auf den Informationen der aktuellen Konfiguration und den Informationen aus den *CPPM-Modellen*. Infolgedessen kann das CPPS-Fähigkeitsmodell

verwendet werden, um die möglichen Produktionssequenzen der aktuellen Konfiguration abzufragen.

- **System Configuration Agents:** Realisieren die *Generierung alternativer Konfigurationen* in Zusammenarbeit mit den *CPPM Agents* durch:
  - die Generierung von möglichen Systemkonfigurationen für die jeweiligen Produktionssequenzen.
  - die Ermittlung von Layoutvarianten für die generierten Systemkonfigurationen.
  - die Ermittlung von Rekonfigurationsaufwänden auf Systemebene, abhängig von den Rekonfigurationsaufwänden der CPPMs auf Maschinenebene (basierend auf den CPPM-Modellen), dem aktuellen Layout und der ermittelten Layoutvariante.
  - die Terminierung von sich selbst, wenn das gewünschte Produkt nicht hergestellt werden kann (also keine gültige Produktionssequenz gefunden werden konnte).
- **CPPM Agents:** Repräsentieren CPPMs und prüfen darüber hinaus, ob das von einem *System Configuration Agent* angeforderte Ausgangsprodukt über einen ihrer Services erreicht werden kann. Zu diesem Zweck haben *CPPM Agents* Kenntnis über ihre alternativen CPPM-Konfigurationen, da sie mehrere VRs sowie die zugehörigen Rekonfigurationsaufwände (auf Maschinenebene) einbeziehen können, um in diese Alternativen zu überführen. So sind sie in der Lage auch Services einzubeziehen, die in einer alternativen CPPM-Konfiguration angeboten werden könnten.
- **Simulation based Optimization (SBO) Agent:** Führt die simulationsbasierte multikriterielle Optimierung der Produktionsaufwände durch die Optimierung der Produktionsparameter (in Bezug auf Zeit, Kosten und Energie) für jeden *System Configuration Agent* durch, der eine umsetzbare Lösung repräsentiert. Der *SBO-Agent* hat die folgenden Aufgaben:
  - Aufbau des Simulationsmodells unter Verwendung der simulationsrelevanten Daten, die vom jeweiligen *System Configuration Agent* bereitgestellt werden.
  - Durchführung der simulationsbasierten multikriteriellen Optimierung unter Berücksichtigung der Steuerungslogik des realen CPPS (d.h. der SBC), die im *Control Model* (4) modelliert ist.
  - Übergabe der Ergebnisse an den *Evaluation Agent*.

Aufgrund der Limitierungen durch die verfügbare Hardware und insbesondere durch die für die SBO verwendete Lizenz beinhaltet die Architektur nur einen SBO-Agenten. In Abhängigkeit von der Verfügbarkeit der Hardware und der Lizenzen (oder einer lizenzunabhängigen Implementierung) ermöglicht der Ansatz jedoch die Einführung zusätzlicher SBO-Agenten und damit eine partielle oder vollständige Parallelisierung. Sobald das Verhältnis 1:1 zwischen *System Configuration Agents* und *SBO Agents* erreicht ist, kann die SBO-Rolle direkt in die *System Configuration Agents* integriert werden, was eine Selbst-Optimierung ohne externe Beteiligung ermöglicht.

- **Evaluation Agent:** Führt die Bewertung der Systemkonfigurationen durch, die sowohl die Rekonfigurations- als auch den Produktionsaufwände berücksichtigt, und ermittelt einen Nutzwert, der darüber hinaus die Gewichtung der gewählten Kriterien (Zeit, Kosten und Energie) berücksichtigt. Nachdem dies für alle gefundenen Systemkonfigurationen durchgeführt wurde, wird die Systemkonfiguration mit dem besten zugewiesenen Nutzwert für den Einsatz ausgewählt.

Das zuvor Beschriebene entspricht im Wesentlichen der Abbildung des MAPE-K-Konzepts. Im Hinblick auf das Rekonfigurationsmanagement eines CPPS gilt folgender Zusammenhang:

- **Monitor:** Überwachung, ob ein Trigger in Form einer Anforderungsänderung (z.B. ein neuer Produktionsauftrag) oder eines Komponentenausfalls auftritt.
- **Analyze:** Durchführung der *Ermittlung des Rekonfigurationsbedarfs*.
- **Plan:** Durchführung der *Rekonfigurationsplanung*, d.h. der *Generierung von alternativen Konfigurationen*, der *Bewertung von Konfigurationen* und der *Auswahl einer Konfiguration*.
- **Execute:** Die *Durchführung von Rekonfigurationsmaßnahmen* erfolgt klassischerweise durch einen manuellen Ansatz. Die Durchführung der Software- und sogar Hardware-Änderungen kann jedoch in Abhängigkeit von den Fähigkeiten der CPPS-Komponenten zur Selbst-(re)konfiguration teilweise oder vollständig autonom erfolgen.
- **Knowledge Components:** Die Wissenskomponenten werden auf verteilte Art und Weise realisiert und befinden sich innerhalb des oben beschriebenen Proxy-Layers. Sie ermöglichen die MAPE-Funktionalität, indem sie das Wissen über das CPPS und seine Komponenten bereitstellen.

Dabei gilt es zu beachten, dass die *Durchführung von Rekonfigurationsmaßnahmen* als optionale Erweiterung des Rekonfigurationsmanagements definiert ist. Durch den vorgestellten dezentralen, parallelisierbaren Ansatz wird also ein autonomes, selbstorganisiertes Rekonfigurationsmanagement erreicht.

Die Steuerung des Produktionssystems wird von einer dedizierten SOA (der SBC) und nicht vom MAS übernommen. Diese *Separation of Concerns* führt zu einer dedizierten, deterministischen Steuerung, die Echtzeitanforderungen erfüllen kann, da die Kommunikation zwischen den beiden Middlewares streng reglementiert ist.

Der Datenzugriff wird durch die Agenten des MAS gekapselt, wo immer Heterogenität auftreten könnte, kann diese durch den Einsatz von Wrappern, wie in Kapitel 2.2 dargelegt wurde, adressiert werden.

Hierzu ist zu sehen, dass die Architektur die CPTMs als statisch gegeben einbindet und daher weder über entsprechende Agenten verfügt, um mit möglicher Heterogenität umzugehen, noch

um mögliche Konfigurationsalternativen aktiv in das Rekonfigurationsmanagement zu integrieren. Dies ist auf das gewählte Realisierungsszenario eines modularen Produktionssystems mit einem festen Matrix-Layout von statisch vorgegebenen Förderbändern zurückzuführen. Das Konzept erlaubt es jedoch problemlos, auch diese Aspekte abzubilden.

Eine weitere Möglichkeit des Einsatzes von Wrapper-Funktionalität innerhalb der Architektur besteht darin, trotz heterogener Serviceschnittstellen das Plug & Produce durch den SOA-basierten Steuerungsansatz zu ermöglichen.

Die vorgeschlagene CPPS-Architektur bietet eine hohe Rekonfigurierbarkeit und Flexibilität durch die Einbeziehung der in Kapitel 2.2 erwähnten Grundprinzipien, wie dem Dekompositionsprinzip und insbesondere der Nutzung der SOA-basierten Steuerung sowie des MAS. Dies resultiert in einer entsprechenden prototypischen Implementierung, die in Kapitel 4 vorgestellt wird und neben der Evaluierung des Konzepts auch zur Überprüfung dieser Behauptung herangezogen werden kann. Während die SOA Rekonfigurierbarkeit und Flexibilität in Bezug auf das zugrundeliegende Produktionssystem und dessen Steuerung bietet, hebt das MAS diese Vorteile auf die konzeptionelle Ebene, d.h. auf die Methodik für das selbstorganisierte Rekonfigurationsmanagement und deren Umsetzung.

## 4 Realisierung des Konzepts und des Tool-Supports

In diesem Kapitel wird die prototypische Implementierung des präsentierten Konzepts beschrieben. Diese bildet wie in Kapitel 1.5 erläutert das zweite Artefakt dieser Forschungsarbeit. Die Realisierung ergibt sich dabei über die Umsetzung der vorgestellten CPPS-Architektur für das selbstorganisierte Rekonfigurationsmanagement. Sie dient als ‚Proof of Concept‘ für die Bereicherung von CPPSs um die Fähigkeit eines selbstorganisierten Rekonfigurationsmanagements. Da die Realisierung des Asset- und des Control-Layers aus Sicht dieser Forschungsarbeit lediglich eine notwendige Basis darstellen und die Modelle des Proxy-Layers bereits in Kapitel 3.3 hinreichend beschrieben wurden, konzentriert sich dieses Kapitel vor allem auf die Realisierung des Management-Layers und dort auch auf die Verwendung der Modelle des Proxy-Layers. Des Weiteren wird in diesem Kapitel auch der Tool-Support aufgezeigt, der realisiert wurde und somit zur Verfügung steht. Im Sinne des DSR wurde die Realisierung, maßgeblich durch studentische Arbeiten, in einem iterativen Prozess ausgeprägt. An vielen Stellen konnten jedoch ältere Varianten von Teilfunktionalitäten als Alternativen beibehalten werden, welche über eine GUI ausgewählt werden können. Der Fokus der Beschreibung der Realisierung liegt jedoch auf der finalen Iteration des Design Zyklus.

### 4.1 Realisierung des Management-Layers

Beginnend mit der obersten Architekturschicht, wurde das Management Layer mit seinen Softwareagenten in Java unter Verwendung des Java Agent Development Framework (JADE) realisiert. Die Implementierung des Multiagentensystems entspricht damit den FIPA-Spezifikationen [156] und nutzt hauptsächlich das Konzept der *Behaviors* und *Messages*, um die in Kapitel 3.4 beschriebenen Abläufe der Softwareagenten zu realisieren. Die Implementierung wurde mit der Eclipse IDE durchgeführt, so dass nun ein neues Framework für das selbstorganisierte Rekonfigurationsmanagement zur Verfügung steht.

Beim Systemstart wird im Rahmen der Initialisierung der Agenten für alle vorliegenden XML-Dateien ein entsprechender CPPM Agent gestartet.

Der **Current Configuration Agent**, verwaltet das *Current Configuration Model* und das *Layout Model*. Im Detail umgesetzt als eine Liste der CPPM-Konfigurationen, einem Layoutgraph (siehe Abbildung 3.5) der das CPPS-Layout abbildet und einem Dictionary, das den CPPM-Konfigurationen eine Position im Layout zuweist. Dabei führt das Hinzufügen, Versetzen (Positionsänderung im Layout) oder Entfernen eines Moduls im Asset-Layer, das in der prototypischen Implementierung durch eine Unity-Simulation realisiert wurde (siehe Kapitel 4.3), zunächst zu einer An- oder Abmeldung beim Discovery-Server des Control-Layers (siehe Kapitel 4.3) bzw. im Falle

des Versetzens eines Moduls im Layout zu einer Aktualisierung der entsprechenden Informationen im Discovery-Server. Die Mitteilung einer solchen Änderung vom Discovery-Server hin zum Current Configuration Agent wurde mit Hilfe eines Mosquitto MQTT-Brokers<sup>19</sup> umgesetzt. Dazu veröffentlicht der Discovery-Server eine Nachricht auf einem Topic des MQTT-Brokers, auf das der Current Configuration Agent abonniert ist (Publish-Subscribe-Prinzip). So kann dem Reconfiguration Demand Agent nach jeder Änderung die aktuelle IST-Konfiguration zur Verfügung gestellt werden.

Anschließend können Trigger zum einen in Form von neuen Produktionsaufträgen, die über den Import-Button der Rekonfigurationsmanagement-GUI (RM-GUI) im XML-Format importiert werden können, in das System gegeben werden (siehe Abbildung 4.1 – (1)). Für eine einfache Kommunikation mit dem restlichen Agentensystem wurde die RM-GUI im Rahmen des sogenannten **GUI Agent** realisiert. Die Produktionsaufträge werden dazu mit Hilfe des in Kapitel 4.2 beschriebenen Toolsupports und entsprechend des in Kapitel 3.3 angeführten Informationsmodells der Produktionsaufträge angelegt. Zum anderen können Trigger in Form eines Komponentenausfalls auf das System gegeben werden. Dies geschieht wie beschrieben mit Hilfe des Discovery-Servers, des MQTT-Brokers und des Current Configuration Agent.

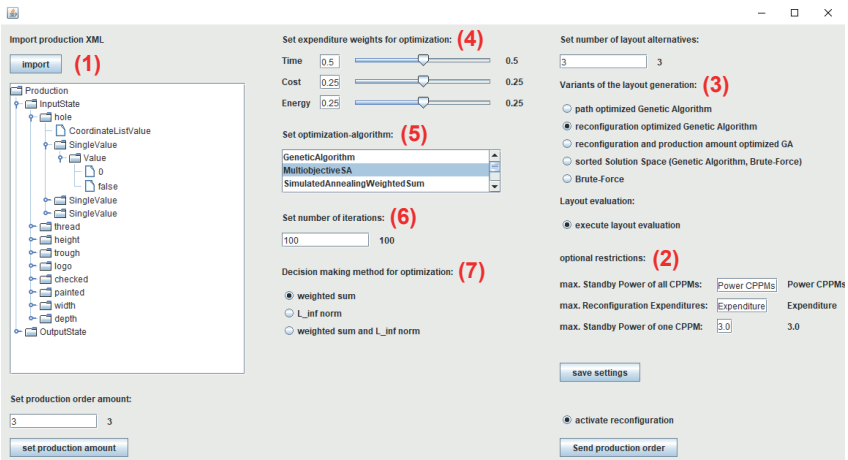


Abbildung 4.1: Rekonfigurationsmanagement-GUI (RM-GUI)

<sup>19</sup> MQTT ist ein weit verbreitetes Netzwerkprotokoll [41] für die Machine-to-Machine-Kommunikation mit dem Vorteil, dass es sehr schlank ist und nur wenig Netzwerkbandbreite benötigt.

Der **Reconfiguration Demand Agent** führt die *Ermittlung des Rekonfigurationsbedarfs* durch, sobald einer der oben genannten Trigger ausgelöst wird. Der Ablauf innerhalb des Reconfiguration Demand Agent ist dazu in Form eines Programmablaufplans in Abbildung 4.2 dargestellt. Mit Hilfe eines implementierten, sogenannten *CyclicBehaviour* wartet der Agent zunächst auf den Eingang einer Nachricht. Dabei kann es sich zum einen um den vom GUI Agent gesendeten Produktionsauftrag, als ein Teil der übermittelten SOLL-Produktion, handeln. Zum anderen kann eine Nachricht die vom Current Configuration Agent übermittelte IST-Konfiguration beinhalten, woraufhin die (Neu-)Bildung des CPPS-Fähigkeitsmodells erfolgt. Hierfür werden die verfügbaren Services, die anhand ihres Prozessoperators beschrieben sind, den CPPM-Konfigurationen der IST-Konfiguration entnommen. Darüber hinaus entspricht das Verfahren zur Bildung des CPPS-Fähigkeitsmodell und auch für die spätere Überprüfung auf Rekonfigurationsbedarf, dem in Kapitel 3.2.1 bereits beschriebenen Konzept. Das CPPS-Fähigkeitsmodell, das die möglichen Produktionssequenzen abbildet, wird dabei als Liste von Graphen gespeichert, die als Wurzel jeweils das Ausgangsprodukt eines Prozessoperators haben. Sind Produktionsauftrag und CPPS-Fähigkeitsmodell vorhanden, so wird die beschriebene Überprüfung auf Rekonfigurationsbedarf durchgeführt. Wird dabei eine valide Produktionssequenz gefunden, so ist keine Rekonfiguration von Nöten und die entsprechende BOP wird gebildet und ausgegeben. Existiert jedoch ein Rekonfigurationsbedarf, so wird basierend auf der aktuellen IST-Konfiguration ein erster System Configuration Agent instanziiert.



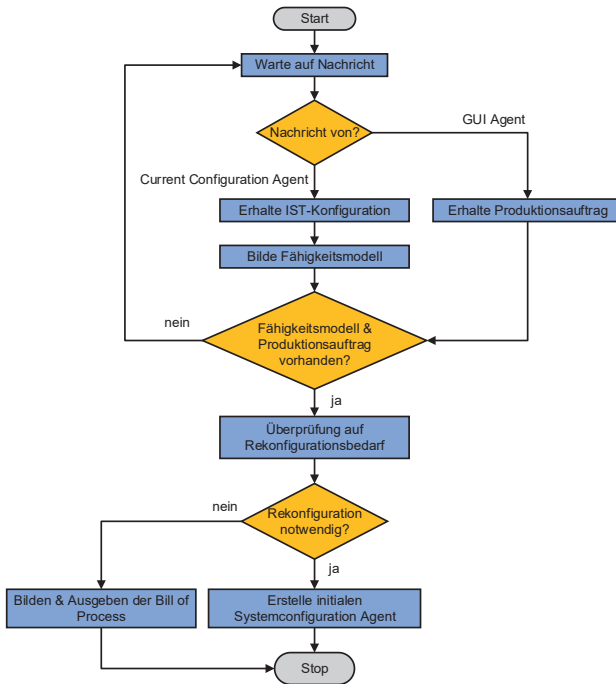


Abbildung 4.2: Programmablaufplan des Reconfiguration Demand Agent

Die **System Configuration Agents** setzen im Zusammenspiel mit den anderen Agenten wesentliche Teile der in Kapitel 3.2.2 beschriebenen *Generierung alternativer Konfigurationen* um, indem sie jeweils eine alternative Systemkonfiguration repräsentieren und diese selbstorganisiert ausprägen. Der Ablauf dem ein jeder System Configuration Agent hierzu folgt, ist in Abbildung 4.3 dargestellt. Bei der Initialisierung wird dem System Configuration Agent zunächst die aktuelle IST-Konfiguration des CPPS sowie die SOLL-Produktion, welche insbesondere den Produktionsauftrag mit dem benötigten Ausgangsprodukt, aber auch die Gewichtung der Kriterien (Zeit, Kosten, Energie) und die weiteren über die RM-GUI eingegebenen Informationen beinhaltet, übergeben. Außerdem erhält der System Configuration Agent auch die bisher festgelegte Produktionssequenz und zudem gegebenenfalls die festgelegte Layoutvariante sofern diese bereits existieren (bei der Erstellung des ersten System Configuration Agent existieren diese noch nicht). Ist bereits eine Layoutvariante für diese Systemkonfiguration vorhanden, so erfolgt als nächster Schritt die Berechnung der Rekonfigurationsaufwände auf Systemebene, welcher später beschrieben wird. Andernfalls wird zunächst überprüft ob bereits eine festgelegte Produktionssequenz vorhanden ist.

Sollte keine bereits festgelegte Produktionssequenz existieren, so werden alle CPPM Agenten angefragt, ob diese in ihrer aktuellen oder einer alternativen Konfiguration einen Service anbieten können, um ein Produkt in das gewünschte Ausgangsprodukt zu transformieren. Als Antwort erhält der System Configuration Agent von den einzelnen CPPM Agenten jeweils eine Liste von geeigneten Services sowie die jeweils dazugehörigen CPPM-Konfigurationen und Rekonfigurationsaufwände auf Maschinenebene. Anschließend wird über alle erhaltenen Services der CPPM Agenten iteriert, wobei für jeden dieser Services ein neuer System Configuration Agent instanziiert wird und diesem die SOLL-Produktion, die IST-Konfiguration, sowie die neu festgelegte Produktionssequenz mit dem aus dem hinzugefügten Service resultierenden neuen, benötigten Ausgangsprodukt übergeben wird (vgl. Kapitel 3.2.2). Nachdem über den letzten der erhaltenen Services iteriert wurde, oder falls keine geeigneten Services zurückgemeldet wurden, beendet sich der System Configuration Agent selbst.

Existiert jedoch bereits eine festgelegte Produktionssequenz, so wird anschließend das entsprechende Fähigkeitsmodell für die betrachtete alternative Systemkonfiguration gebildet. Der Unterschied zur Bildung des Fähigkeitsmodells durch den Reconfiguration Demand Agent besteht darin, dass hier, entsprechend dem in Kapitel 3.2.2 beschriebenen Konzept lediglich ein einziger Pfad der Baumstruktur besteht. Anschließend wird auch dieses Fähigkeitsmodell der alternativen Systemkonfiguration dazu verwendet, um nach dem bereits beschriebenen Verfahren zu prüfen, ob eine valide Produktionssequenz existiert. An dieser Stelle reduziert sich dieses Verfahren jedoch auf die Betrachtung einer einzigen, festgelegten Produktionssequenz.

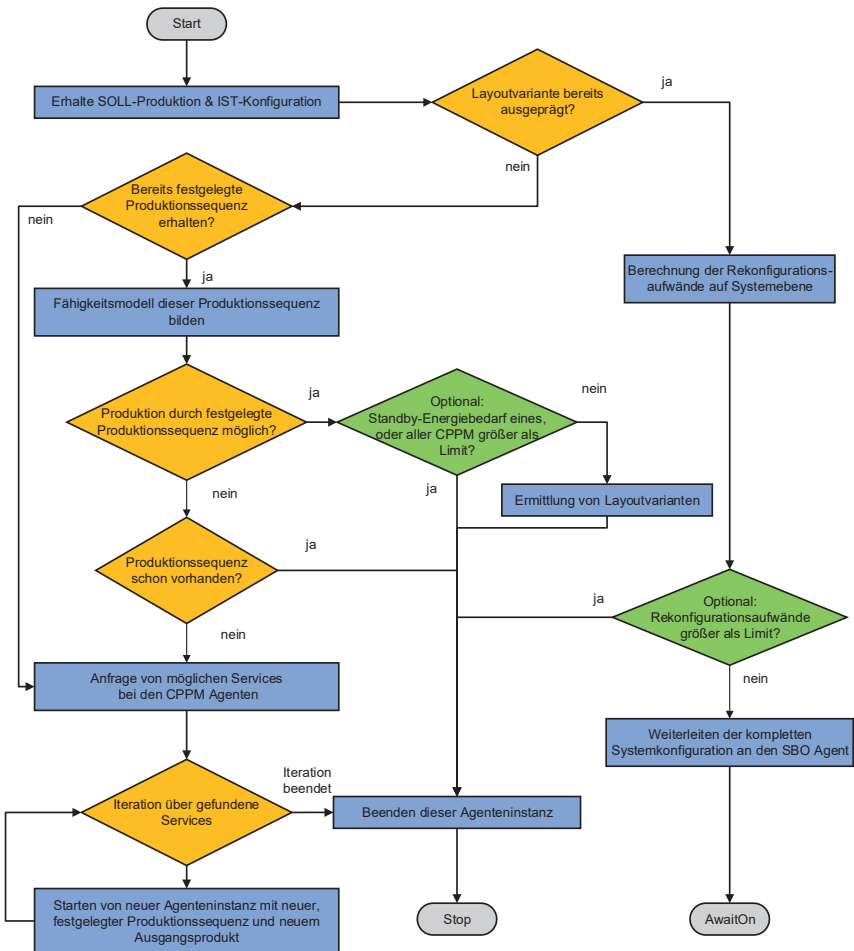


Abbildung 4.3: Programmablaufplan eines System Configuration Agent

Ist eine Produktion mit der festgelegten Produktionssequenz nicht möglich, so erfolgt zunächst eine Schleifenerkennung indem überprüft wird, ob dieser Produktionsauftrag mit der festgelegten Produktionssequenz schon einmal betrachtet wurde. Ist dies der Fall, so wird dieser Agent beendet, ist dies nicht der Fall, so erfolgt die bereits beschriebene Anfrage an alle CPPM Agenten nach geeigneten Services, allerdings ergänzt um die bestehende Produktionssequenz damit diese entsprechend erweitert werden kann.

Ist jedoch eine Produktion mit der festgelegten Produktionssequenz möglich, so wird je nach Auswahl über die RM-GUI (siehe Abbildung 4.1 – (2)) zunächst geprüft, ob dort gesetzte Grenzwerte für den Standby-Energieverbrauch eines einzelnen CPPM oder aller CPPMs in Summe, für die betrachtete Systemkonfigurationsalternative überschritten werden. Dazu werden die Informationen aus den CPPM-Modellen verwendet und im Falle einer Grenzwertüberschreitung beendet sich der System Configuration Agent selbst.

Andernfalls beginnt die Ermittlung von Layoutvarianten. Diese kann entsprechend dem in Kapitel 3.2.2 beschriebenen Konzept, neben der Verwendung eines Brute-Force-Ansatzes um alle Layoutvarianten zu erzeugen, layoutoptimiert oder rekonfigurationsoptimiert erfolgen. Die Auswahl erfolgt über die RM-GUI (siehe Abbildung 4.1 – (3)). Dabei können sowohl layoutoptimierte als auch rekonfigurationsoptimierte Layoutvarianten mit Hilfe der beiden Verfahrensvarianten eines Genetischen Algorithmus (GA) ermittelt werden. Grundsätzlich gilt für die GA-basierte Layoutvariantenfindung, dass dem GA entsprechend der jeweiligen Systemkonfiguration die Anzahl der CPPMs sowie die Produktionssequenz, also Ausführungsreihenfolge der Services dieser CPPMs, übergeben wird. In diesem Kontext stellt ein Individuum eine Layoutvariante dar, Abbildung 4.4 gibt einen Überblick über diesen Zusammenhang. Ein Gen gibt die Zuweisung der Positionsnummer im Layout zu einem Serviceschritt entsprechend der Servicereihenfolge an, welcher von dem zugehörigen CPPM in der geplanten CPPM-Konfiguration angeboten wird. Die Positionsnummern entsprechen dabei den möglichen Standorten für CPPMs im Matrix-Layout des Produktionssystems, wie bereits in Abbildung 3.5 dargestellt. Ein Allel, also die spezifische Ausprägung eines Gens, kann für dieses Layout Werte zwischen 1 und 12 annehmen, da die Positionen 0 und 13 der Source bzw. Sink für den Ein- bzw. Austritt der Produkte vorbehalten sind. Um die Verwendung unzulässiger Positionsnummern sowie eine zusätzliche Dekodierung zu vermeiden, werden reelle Zahlen verwendet und auf eine Kodierung verzichtet.

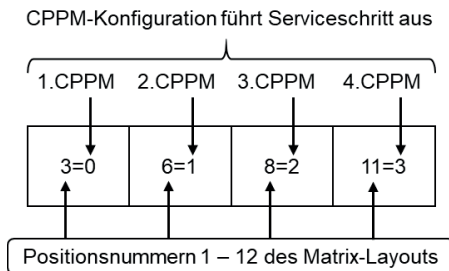


Abbildung 4.4: Darstellung eines Individuums (Layoutvariante)

Eine vereinfachte Übersicht der konkreten Ausprägung beider Verfahrensvarianten des GA ist in Abbildung 4.5 dargestellt und zeigt insbesondere die jeweils eingesetzten Operatoren. Dabei zielt das layoutoptimierte Verfahren auf einen minimalen Transportweg ab, ohne die aktuelle Position

von CPPMs der aktuell eingesetzten IST-Konfiguration zu berücksichtigen. Die Ausgangspopulation wird daher zufällig, basierend auf den für die betrachtete Systemkonfiguration verfügbaren CPPMs gebildet. Das rekonfigurationsoptimierte Verfahren hingegen zielt auf die Minimierung der Rekonfigurationsaufwände ab und erlaubt in seiner ursprünglichen Form keine Positionsänderung bereits im Layout verorteter CPPMs. Sie werden daher in diesem Kontext auch als *immobile CPPMs* bezeichnet und in den Operationen Kreuzung, Mutation sowie bei der Generierung der Ausgangspopulation als solche behandelt (siehe Abbildung 4.5 orange Markierungen). Über eine weitere Option in der RM-GUI (siehe Abbildung 4.1 – (3)) ist es jedoch möglich, einzelne immobile CPPMs in Abhängigkeit des Produktionsvolumens (d.h. der geforderten Stückzahl) dennoch zu versetzen, um Kompromisslösungen hinsichtlich des Rekonfigurations- und Transportaufwands zu erzeugen. Dieses Sub-Verfahren basiert auf einer komplexen und experimentellen Ausprägung eines sogenannten *Educated Guess*, um den entstehenden Umweg durch die Behandlung eines CPPMs als immobil gegenüber dem De- und Installationsaufwand durch das Versetzen dieses CPPMs abzuschätzen. Dies geschieht unter Verwendung des Dijkstra-Algorithmus zur Wegfindung sowie der geforderten Stückzahl und wird aus Gründen der Übersichtlichkeit nicht näher beschrieben. Unabhängig von der gewählten Verfahrensvariante schließt sich eine Fitnessbewertung der Ausgangspopulation an. Zur besseren Übersicht, sind identisch ausgeprägte Schritte beider Verfahrensvarianten in Abbildung 4.5 türkis dargestellt. Die Fitnessbewertung erfolgt anhand des Transportwegs, den ein Produkt für die jeweilige Layoutvariante zurücklegen muss, mit Hilfe des Dijkstra-Algorithmus. Hierfür wird das in Abbildung 3.5 dargestellte Matrix-Layout des Produktionssystems als ungerichteter Graph abgebildet. Diese vereinfachende Betrachtung des Transportwegs mit Hilfe des Dijkstra-Algorithmus, begründet sich darin, dass im Zuge dieser Ermittlung von Layoutvarianten eine möglichst sinnvolle frühzeitige Reduktion des Lösungsraumes ermöglicht werden soll. Eine multikriterielle Betrachtung des tatsächlichen Materialflusses unter Berücksichtigung des eingesetzten Steuerungsverfahrens und der gewählten Parametrierung erfolgt erst im Zuge der späteren simulationsbasierten Optimierung. Um also eine vollständige Betrachtung des Lösungsraums zu gewährleisten, kann der Brute-Force-Ansatz für die Ermittlung der Layoutvarianten gewählt werden.

Die layoutoptimierte Verfahrensvariante setzt als Selektionsmethode die *Roulette-Wheel Selection* ein um dem Konflikt zwischen Diversität und Feinabstimmung Rechnung zu tragen, dabei werden starke Individuen begünstigt. Hinsichtlich der Kreuzung wird ein *One-Point Crossover* eingesetzt, da durch die geringe Anzahl an Kreuzungspunkten die einzuhaltende Serviceabfolge eher erhalten bleibt. Für die Mutation wird ein Serviceschritt zufällig ausgewählt und dessen zugehörige Position mutiert, wobei alle Positionen des Layouts (Positionen 1-12) in Betracht gezogen werden. Unter der Annahme das eine tendenziell homogene Population erzeugt wird, kommt für die layoutoptimierte Verfahrensvariante das *General Replacement* als Ersetzungsschema zum Einsatz.

Die rekonfigurationsoptimierte Verfahrensvariante setzt hingegen die *Tournament Selection*, mit einer selbstadaptiven Anzahl an Wettkampfteilnehmern, zur Selektion ein. In Anbetracht der immobilen CPPMs kommt das *Uniform Crossover* zum Einsatz, um auch unter Beibehaltung bestimmter Positionen eine Variation der Individuen zu erreichen. Im Gegensatz zur Mutation der layoutoptimierten Verfahrensvariante werden hier die Serviceschritte und Positionen von immobilen CPPMs bei der Mutation ausgeschlossen. Aufgrund der Annahme, dass durch das Uniform Crossover eine tendenziell heterogene Population erzeugt wird, verwendet diese Verfahrensvariante das *Delete-n-Last* als Ersetzungsschema.

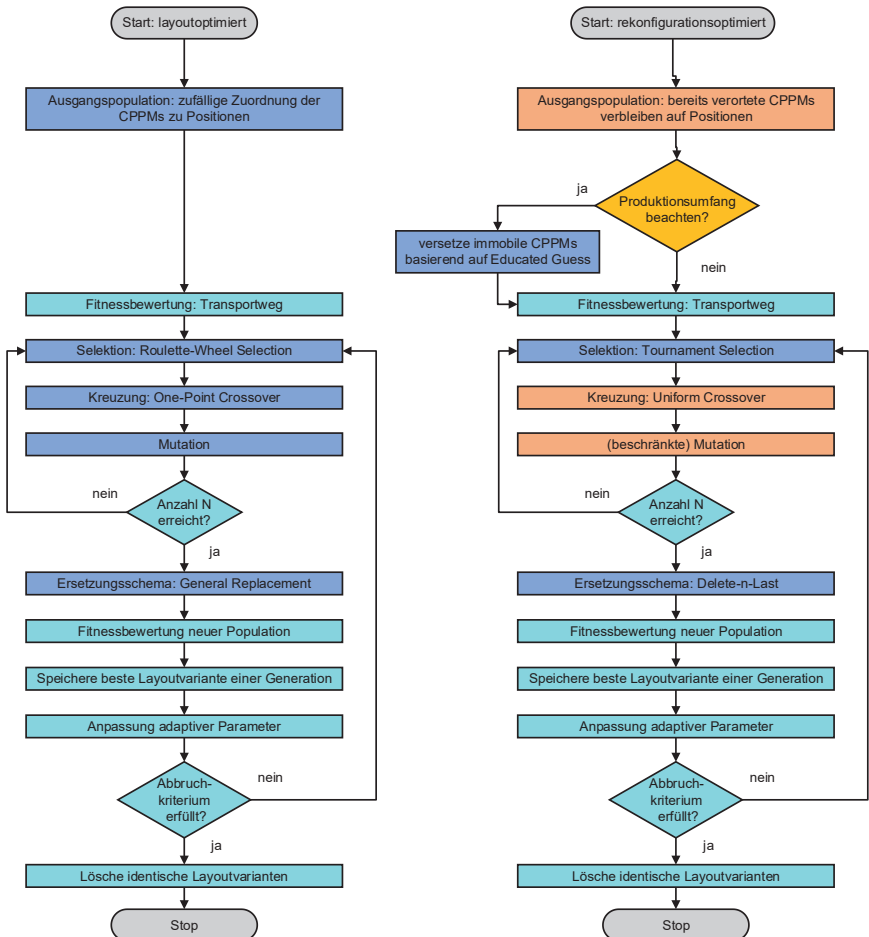


Abbildung 4.5: Vereinfachte Übersicht der Ermittlung von Layoutvarianten

Für beide Verfahrensvarianten ist festzuhalten, dass die Operationen Selektion, Kreuzung und Mutation iterativ durchgeführt werden, bis  $N$  neue Individuen erzeugt wurden, um die alte Population entsprechend ersetzen zu können. Nach Anwendung des beschriebenen spezifischen Ersetzungsschemas erfolgt die Bewertung der Fitness aller Individuen der neuen Population, diese erfolgt ebenso wie die oben angeführte Fitnessbewertung der Ausgangspopulation anhand des Transportwegs und mit Hilfe des Dijkstra-Algorithmus. Darüber hinaus wird für jede Generation die Layoutvariante mit dem besten Fitnesswert gespeichert. Sollte der Fitnesswert der vorherigen und der nachfolgenden Generation denselben Wert aufweisen, so werden deren Gene verglichen und beide Varianten werden beibehalten sofern sie unterschiedliche Gene aufweisen. Anschließend erfolgt für beide Verfahrensvarianten die Anpassung der selbstadaptiven Parameter, dabei wird zu Beginn eine hohe Kreuzungswahrscheinlichkeit genutzt welche nach jeweils 25% der Iterationen des GA um 10% reduziert wird um das Maß an Variation zu regulieren. Zudem wird die Mutationswahrscheinlichkeit jeweils um 5% reduziert. Bei der rekonfigurationsoptimierten Verfahrensvariante wird an dieser Stelle darüber hinaus die Anzahl der Wettkampfteilnehmer der Tournament Selection um zwei erhöht. Danach erfolgt die Überprüfung des Abbruchkriteriums, das durch eine maximale Anzahl von Iterationen umgesetzt ist. Nach Erfüllung des Abbruchkriteriums werden identische Layoutvarianten gelöscht. Die für die Realisierung der GA-basierten Verfahrensvarianten gewählten Parameter hängen insbesondere von der Anzahl der eingesetzten CPPMs ab und sind in Tabelle B.1 des Anhangs aufgeführt. Zuletzt erfolgt die Zusammenführung der besten Layoutvarianten der gewählten Verfahrensvarianten.

Zum Abschluss der Ermittlung von Layoutvarianten wird für jede gefundene Layoutvariante ein neuer System Configuration Agent erzeugt, welcher die bisher bestehenden Informationen zu der betrachteten Systemkonfiguration ergänzt um die jeweilige Layoutvariante erhält. Der ursprüngliche System Configuration Agent beendet sich anschließend selbst.

Sofern ein System Configuration Agent, wie eingangs beschrieben, bei seiner Initialisierung eine Layoutvariante erhält, erfolgt die Berechnung der kriterienspezifischen Rekonfigurationsaufwände auf Systemebene  $a_{z, \text{reconfiguration}}$ . Dazu werden einerseits die Rekonfigurationsaufwände auf Modulebene aufsummiert, die von den CPPM Agenten geliefert werden, wenn diese einem System Configuration Agent auf die Anfrage nach geeigneten Services antworten. Zum anderen werden die Installations- und Deinstallationsaufwände berechnet und aufsummiert, die für das Hinzufügen und Entfernen von Produktionsmodule anfallen, wenn die IST-Konfiguration in die betrachtete neue Layoutvariante überführt wird. Die Berechnung der Rekonfigurationsaufwände  $a_{z, \text{reconfiguration}}$  für jedes Bewertungskriterium  $z$  erfolgt also wie bereits im Konzept beschrieben (siehe Kapitel 3) und nutzt die notwendigen Informationen des Proxy-Layers. An dieser Stelle sei auf den Anhang verwiesen, der einen noch detaillierteren Überblick über die Modellierung gibt.

Als ein Beispiel für eine weitere optionale Filterstufe wurde implementiert, dass anschließend, sofern diese Option über die RM-GUI (siehe Abbildung 4.1 – (2)) gewählt wurde, überprüft wird ob der zeitliche Rekonfigurationsaufwand einen einstellbaren Grenzwert überschreitet. Ist dies der Fall, so beendet sich der System Configuration Agent selbst.

Andernfalls erfolgt schließlich die Weiterleitung der ausgeprägten, alternativen Systemkonfiguration an den SBO Agent und der System Configuration Agent wartet im Zustand *AwaitOn* auf den Eingang weiterer Nachrichten.

Bei der Initialisierung jedes **CPPM Agents** wird diesem sowohl die Benennung der entsprechenden XML-Datei mitgeteilt, die Informationen über das Produktionsmodul enthält, das von diesem CPPM Agenten repräsentiert werden soll, als auch seine initiale CPPM-Konfiguration. Anhand der Benennung der XML-Datei liest der CPPM Agent diese zunächst aus, um die in Kapitel 3.3 beschriebenen Informationen zu erhalten (siehe Abbildung 3.8). Der konkrete Workflow zur Erstellung der XML-Dateien basierend auf dem in Kapitel 3.3 definierten Informationsmodells (siehe Abbildung 3.8) bis hin zur Integration der Informationen in das CPPS wird in Kapitel 4.2 erläutert. Auf diese Weise steht dem CPPM Agent fortan sein CPPM-Modell mit der Spezifikation des Produktionsmoduls zur Verfügung, die insbesondere dessen Modulkonfigurationsalternativen mit ihren entsprechenden Services beschreibt. Danach wartet der CPPM Agent auf die oben beschriebene Anfrage von möglichen Services durch einen System Configuration Agent. Sobald eine solche Anfrage eingeht, prüft der CPPM Agent zunächst, ob sein CPPM bereits in der betrachteten Systemkonfiguration eingeplant wurde. Ist dies der Fall, werden im Folgenden nur die Services berücksichtigt, die von der eingeplanten Modulkonfiguration angeboten werden. Anschließend wird über alle Modulkonfigurationen und deren Services iteriert und geprüft, ob einer dieser Services eine Transformation in das vom System Configuration Agent geforderte Ausgangsprodukt realisieren kann. Dies geschieht durch einen Vergleich der Merkmale<sup>20</sup>, die den Ausgangszustand eines Produktes bei Ausführung des jeweiligen Services beschreiben, mit den Merkmalen des geforderten Ausgangsproduktes (vgl. Kapitel 3.2.2 und Kapitel 3.3). Dabei wird auch geprüft, ob der betrachtete Service für die Verarbeitung des Produktmaterials geeignet ist. Die dazu benötigte Beschreibung der Services anhand ihrer Merkmale und ihrer Eignung für Materialien wird dabei aus dem jeweiligen CPPM-Modell entnommen. Für alle so als geeignet identifizierten Services werden die Rekonfigurationsaufwände für die Rekonfiguration auf Maschinenebene, also für die Überführung von der aktuellen in die entsprechende Modulkonfiguration, ermittelt. Hierzu werden bereits bei der Initialisierung des CPPM Agenten die Installations- und Deinstallationsaufwände der möglichen Komponenten des CPPM, sowie die Komponentenlisten

---

<sup>20</sup> Im Detail werden dabei auch Untermerkmale berücksichtigt, in Kapitel 3.3 und Abbildung 3.8 ist deren Modellierung bereits beschrieben.



aller Modulkonfigurationen des CPPM aus der zugehörigen XML-Datei ausgelesen. Basierend auf diesen Informationen werden die Rekonfigurationsaufwände für die Überführung der verschiedenen Modulkonfigurationen ineinander in Abhängigkeit der dafür zu installierenden und zu deinstallierenden Komponenten berechnet. In Anlehnung an [12] werden die Ergebnisse in Rekonfigurationsmatrizen<sup>21</sup> abgelegt, die zu diesem Zeitpunkt genutzt werden können um die Rekonfigurationsaufwände zu erhalten. Abschließend wird dem anfragenden System Configuration Agent eine Liste der geeigneten Services sowie die zugehörigen CPPM-Konfigurationen und Rekonfigurationsaufwände auf Maschinenebene zurückgemeldet. Danach wartet der CPPM Agent auf die nächste Anfrage eines System Configuration Agenten.

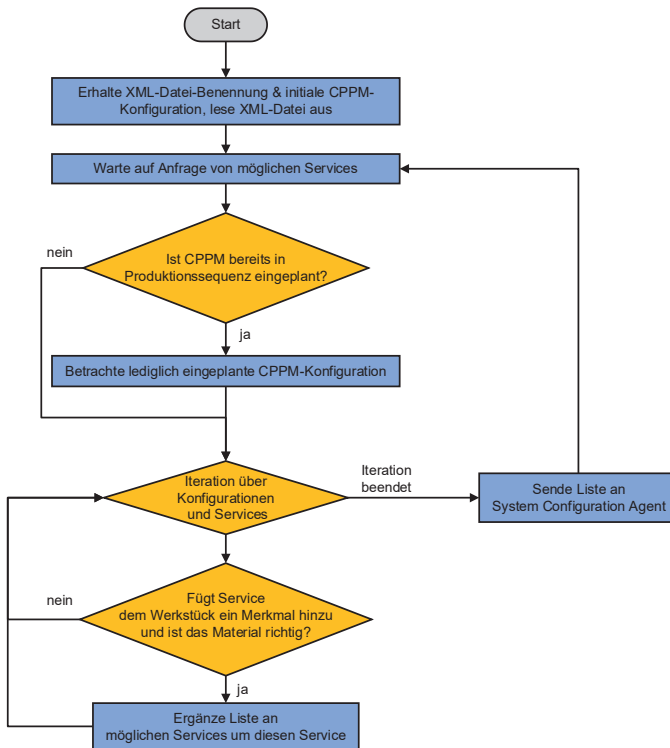


Abbildung 4.6: Programmablaufplan der CPPM Agenten

<sup>21</sup> Ein Beispiel hierfür ist in Tabelle B.2 des Anhangs angeführt.

Der **SBO Agent** realisiert, wie in Kapitel 3.2.2 und Kapitel 3.4 beschrieben, die simulationsbasierte multikriterielle Optimierung der Produktionsaufwände, indem er für jede durch einen System Configuration Agent ausgeprägte, alternative Systemkonfiguration die Optimierung der Produktionsparameter (hinsichtlich Zeit, Kosten und Energie) durchführt. Der hierfür implementierte Ablauf des SBO Agent ist in Abbildung 4.7 dargestellt.

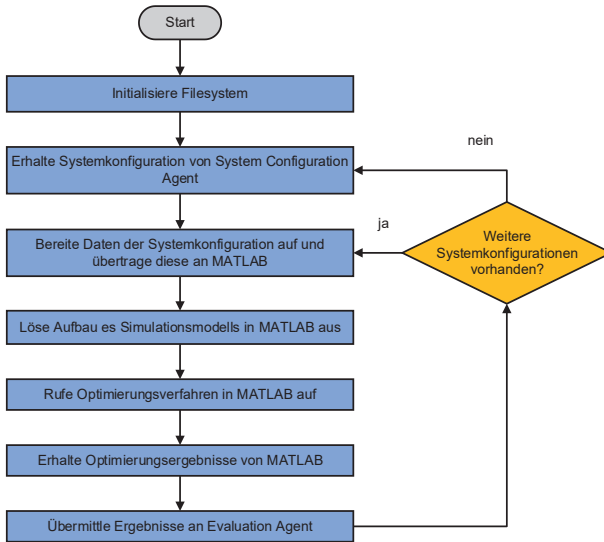


Abbildung 4.7: Programmablaufplan des SBO Agent

Der SBO Agent wird bei der Initialisierung des Agentensystems gestartet und erstellt zunächst einen Ordner in dem später der MATLAB Workspace und das zu erstellende Simulationsmodell für jede zu betrachtende Systemkonfiguration abgelegt werden können. Danach ist der SBO Agent bereit, Nachrichten von den System Configuration Agenten zu empfangen, diese werden in einer Message Queue zwischen gepuffert und können so vom SBO Agent über ein *CyclicBehaviour* sequentiell abgearbeitet werden. Sobald mindestens eine Nachricht empfangen wurde, beginnt der SBO Agent, mit Hilfe eines *OneShotBehaviours*, mit dem Aufbau des Simulationsmodells für die betrachtete Systemkonfiguration. Dazu werden die erhaltenen Daten der jeweiligen Systemkonfiguration zunächst in Strukturen überführt, die MATLAB übergeben werden können, beispielsweise MATLAB Structures oder einfache Datentypen wie Integer. Insbesondere handelt es sich dabei um Informationen über die eingesetzten CPPM-Konfigurationen aus deren CPPM-Modellen (siehe auch Kapitel 3.3), wie den Formeln zur Berechnung der Produktionsaufwände in Ab-

hängigkeit der Produktionsparameter einschließlich der möglichen Wertebereiche und Schrittweiten der Produktionsparameter die es zu optimieren gilt. Zudem umfasst dies die über die RM-GUI (siehe Abbildung 4.1 – (4)) eingestellte Gewichtung der Kriterien, das gewählte Optimierungsverfahren und die geforderte Stückzahl, sowie darüber hinaus die Positionen der CPPMs im Produktionslayout und die Ausführungsreihenfolge der Services. Anschließend verwendet der SBO Agent die *MATLAB Engine API for Java*, um gemäß Abbildung 4.8 die Daten an MATLAB zu übergeben, das Simulationsmodell zu erstellen und schließlich die simulationsbasierte Optimierung durchzuführen. Ein Übersichtsbild des MATLAB Workspace für eine betrachtete Systemkonfiguration nach der initialen Datenübertragung ist im Anhang in Abbildung B.5 aufgeführt.

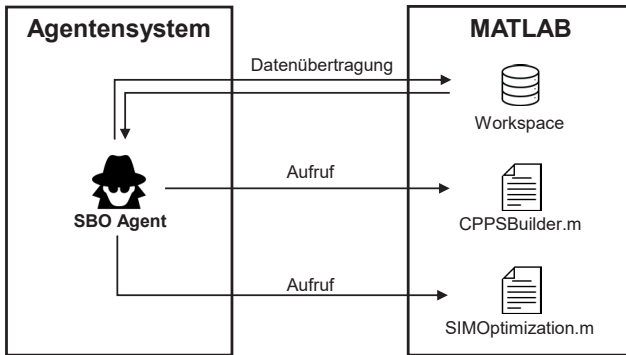


Abbildung 4.8: Kommunikation zwischen SBO Agent und MATLAB

Jede gefundene Systemkonfiguration wird in *MATLAB Simulink* unter Verwendung von *Stateflow* erstellt und optimiert, wodurch die Modellierung sowie eine diskrete ereignisorientierte Simulation basierend auf parallel ausgeführten Zustandsautomaten ermöglicht wird. Der Aufbau des Simulationsmodells wird dabei ausgelöst indem der SBO Agent das MATLAB Skript *CPPSBuilder.m* aufruft und so das in Abbildung 4.9 dargestellte Simulink Modell in Form eines *Discrete Event Charts* erzeugt wird. Die Ausgangssignale des Simulink Modells, also die Produktionsaufwände im Sinne der für die Fertigstellung des Auftrags benötigte Zeit, Kosten und Energie, werden an dieser Stelle über die drei Blöcke rechts im Bild an den MATLAB Workspace herausgeführt. Sie entsprechen den Ausgangssignalen  $F(x)$ , deren Bildung bereits in Kapitel 3.2.2 des Konzepts beschrieben wurde. Das Simulationsmodell ersetzt somit, wie bereits im Konzept (Kapitel 3.2.2) beschrieben, eine mathematische Formulierung der Zielfunktionen für die spätere Optimierung.

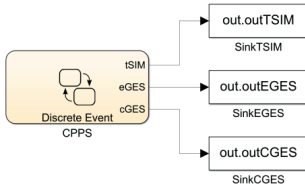


Abbildung 4.9: Simulink Modell einer Systemkonfiguration des CPPS

Innerhalb des *Discrete Event Charts* werden wie in Abbildung 4.10 dargestellt, die jeweiligen Systemkonfigurationen und ihr Layout durch die Zustandsautomaten der zugehörigen CPTMs und CPPMs modelliert. Darüber hinaus werden die Produkte zunächst im Generator-Block (Source-Modul) erzeugt, dann als sogenannte *Messages* von den CPTM- und CPPM-Blöcken verarbeitet, um die erforderlichen Produktionssequenzen entsprechend der Steuerungslogik der SBC abarbeiten zu können, und schlussendlich durch den Terminator-Block (Sink-Modul) eliminiert<sup>22</sup>. Die verschiedenen Blöcke werden dabei als parallel laufende Subcharts erzeugt. Die CPM Blöcke nutzen dabei die zuvor an den MATLAB Workspace übertragenen Formeln zur Berechnung der Produktionsaufwände in Abhängigkeit der Produktionsparameter um die beschriebene Bildung der Ausgangssignale zu realisieren. Basierend auf den allgemein gültigen Formeln (1) – (5) aus Kapitel 3.2.2 gilt für das Ausgangssignal *eGES* also für den Gesamtenergieaufwand  $E_{ges}$ :

$$E_{ges} = \sum_{n=1}^{n_{max}} E_{n,CPM} \quad (9)$$

$$E_{n,CPM} = E_{n,standby} + E_{n,service} \quad (10)$$

$$E_{n,standby} = \sum_{k=1}^s e_{n,k,standby} = \sum_{k=1}^s p_{n,k,standby} * t_{n,k,standby} \quad (11)$$

$$E_{n,service} = \sum_{k=1}^p e_{n,k,service} = \sum_{k=1}^p p_{n,k,service} * t_{n,k,service} \quad (12)$$

Der Gesamtenergieaufwand  $E_{ges}$  für die Abarbeitung der betrachteten Produktion ergibt sich also nach (9) aus den Gesamtenergieaufwänden aller CPMs, welche sich nach (10) jeweils aus deren Standby- und Service-Energieaufwänden ergeben. Diese ergeben sich nach (11) bzw. (12) wiederum über die Multiplikation der Leistungsaufnahme  $p$  eines CPMs  $n$  mit der Verweildauer  $t$  im

<sup>22</sup> Die konkrete Ausprägung der vier genannten Blocktypen als Subcharts ist im Anhang der Realisierung dargestellt.

Standby- bzw. Service-Zustand über alle auftretenden Standby-Zyklen  $s$  bzw. Service-Zyklen  $p$ <sup>23</sup>. Da sich die Produktionskosten im realisierten Prototyp über das Produkt aus Strompreis  $c$  und dem Energieaufwand berechnet, gilt für die Bildung des Ausgangssignals  $cGES$  also für die Gesamtkosten  $K_{ges}$  entsprechend:

$$K_{ges} = \sum_{n=1}^{n_{max}} K_{n,CPM} \quad (13)$$

$$K_{n,CPM} = K_{n,standby} + K_{n,service} \quad (14)$$

$$K_{n,standby} = \sum_{k=1}^s k_{n,k,standby} = \sum_{k=1}^s c * e_{n,k,standby} \quad (15)$$

$$K_{n,service} = \sum_{k=1}^p k_{n,k,service} = \sum_{k=1}^p c * e_{n,k,service} \quad (16)$$

Über das Ausgangssignal tSIM wird die Zeit für die Fertigstellung des Auftrags geliefert, welche sich nach Formel (5) aus Kapitel 3.2.2 über die Summe aller Standby- und Servicezykluszeiten eines einzelnen CPM berechnen lässt.

Der zulässige Bereich der Produktionsparameter jedes CPPMs, welche die Entwurfsvariablen (engl. Design Values) der nachfolgenden Optimierung darstellen, wird dabei durch die Wertebereiche und Schrittweiten der Produktionsparameter definiert<sup>24</sup>. Die Nachbildung der Steuerungslogik der SBC (siehe auch Kapitel 4.3) wurde als MATLAB-Funktion realisiert, die nach Ausführung jedes Services ausgeführt wird, um das nächste CPM für das Produkt zu ermitteln. Hierzu nutzt sie die aktuelle Position des Produkts, den als nächstes benötigten Fertigungsservice entsprechend der Ausführungsreihenfolge der Services, die Position der vorhandenen CPPMs und weist welche Fertigungsservice diese anbieten. Zudem ermittelt sie mit Hilfe des Dijkstra-Algorithmus und basierend auf der Verfügbarkeit der CPTMs und dem Layout-Graph den kürzesten Weg. Anschließend werden die benötigten CPTM für dieses Produkt reserviert und das Produkt wird transportiert, wobei nach der Nutzung eines CPTM (bzw. von dessen Transportservice) dieses wieder freigeben wird und somit weiteren Produkten zur Verfügung steht. Hierzu wird der erweiterte Layout-Graph, welcher die Positionen aller CPTM, also bi-direktionaler und omni-direktionaler Logistikmodule (siehe auch Abbildung 3.7 im Vergleich zu Abbildung 3.5) durch die

---

<sup>23</sup> Die konkrete Berechnung am Beispiel eines Bohrmoduls ist im Anhang der Realisierung beschrieben.

<sup>24</sup> Für ein besseres Verständnis des im Anhang in Abbildung B.5 aufgeführten MATLAB Workspace sei auf die Verwendung der Abkürzung DV für Design Values hingewiesen.



Da im Sinne des DSR, insbesondere getrieben durch mehrere studentische Arbeiten, mehrere Iterationen des Design Zyklus zur Entwicklung dieses Artefaktes durchlaufen wurden, wurden mehrere Optimierungsverfahren realisiert. Die Optimierungsverfahren aus den frühen Iterationen wurden jedoch nicht verworfen, stattdessen bietet der realisierte Prototyp nun mehrere unabhängige Optimierungsverfahren zur Auswahl über die RM-GUI (siehe Abbildung 4.1 – (5)) an, welche dann im Zuge der Ausführung des oben genannten MATLAB Skripts *SIMOptimization.m* genutzt werden. Die aktuelle Implementierung erlaubt es, neben der bereits in [32] publiziert und beschriebenen Methode zur Lösung eines einkriteriellen Ersatzproblems insbesondere ein multikriterielles Simulated Annealing oder einen multikriteriellen Genetischen Algorithmus einzusetzen, welche in der letzten Iteration des Design Zyklus realisiert wurden. Des Weiteren stehen eine auf der MATLAB Optimization Toolbox basierende Variante des Simulated Annealing sowie des Genetischen Algorithmus aus früheren Iterationen zur Verfügung, die jedoch durch die Varianten der finalen Iteration abgelöst wurden. Außerdem kann ein Monte Carlo Algorithmus als Optimierungsverfahren gewählt werden. Zwei Beispiele für eine laufende Optimierung mit den Toolbox-basierten Varianten des Simulated Annealing und des Genetischen Algorithmus sind in Abbildung 4.11 und Abbildung 4.12 dargestellt.

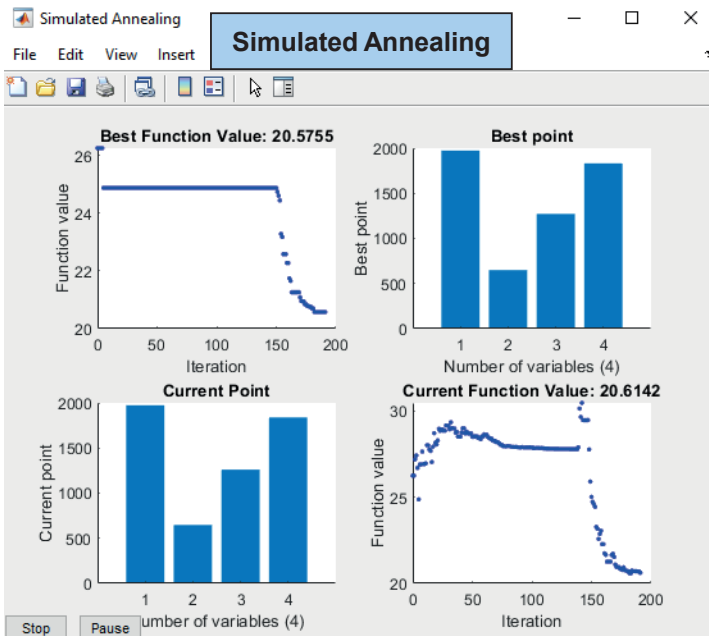


Abbildung 4.11: Laufende Optimierung unter Einsatz eines Toolbox-basierten Simulated Annealing Algorithmus (Beispiel)

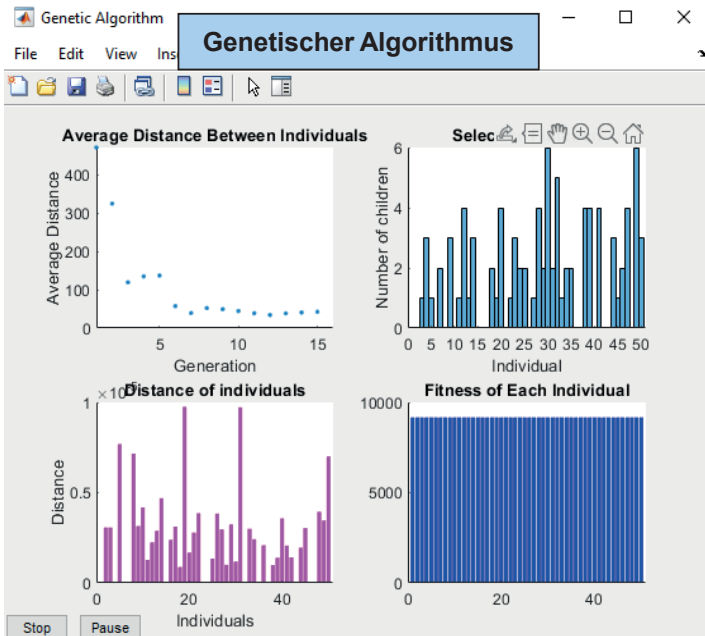


Abbildung 4.12: Laufende Optimierung unter Einsatz eines Toolbox-basierten Genetischen Algorithmus (Beispiel)

Nachfolgend wird exemplarisch die Realisierung des Optimierungsverfahrens auf Basis des in der letzten Iteration des Design Zyklus entwickelten multikriteriellen Simulated Annealing, englisch Multi Objective Simulated Annealing (MOSA), vorgestellt, da es eine gute Performance aufweist und am ausführlichsten evaluiert wurde.

In Abbildung 4.13 ist daher der Ablauf des realisierten MOSA Algorithmus in Form eines Flussdiagramms dargestellt.



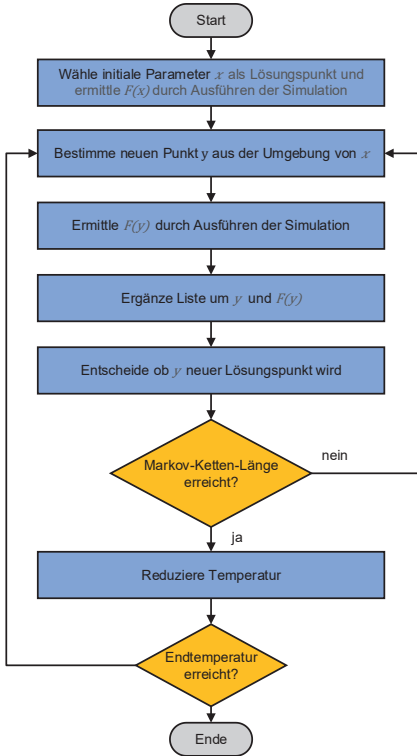


Abbildung 4.13: Übersicht der Optimierung mit dem realisierten MOSA Algorithmus

Hierbei wird zunächst ein initialer Lösungspunkt  $x$  gewählt indem für alle Produktionsparameter die es zu optimieren gilt deren mögliche Wertebereiche und Schrittweiten aus den CPPM-Modellen betrachtet werden. Dadurch wird der zulässige Bereich  $\Omega$  definiert, wobei für jeden Produktionsparameter  $x_i$  der Minimalwert  $x_{i,min}$ , der Maximalwert  $x_{i,max}$  und eine Schrittweite  $x_{i,step}$  gegeben sind und es gilt:

$$\Omega = \{(x_i)_{1 \leq i \leq n} \in \mathbb{R}^n | \exists \gamma \in \mathbb{N}: \forall i: x_{i,min} \leq x_i \leq x_{i,max} \wedge x_i = x_{i,min} + \gamma x_{i,step}\} \quad (17)$$

Der initiale Lösungspunkt  $x$  ergibt sich, indem für jeden zu optimierenden Produktionsparameter der mittlere Wert zwischen  $x_{i,min}$  und  $x_{i,max}$  gewählt wird, welcher unter Berücksichtigung der Schrittweite  $x_{i,step}$  einen zulässigen Wert darstellt. Anschließend wird die Simulation ausgeführt wodurch die zugehörigen Produktionsaufwände  $F(x)$  ermittelt werden. Für jeden Punkt  $x \in \Omega$  ist

seine Umgebung  $\Omega_{umg}(x)$  als diejenigen Punkte definiert welche über einen Schritt in jede Richtung erreichbar sind und es gilt:

$$\Omega_{umg}(x) = \{y \in \Omega \mid \exists \gamma \in \{-1, 0, 1\}^n: y = x + \gamma \odot x_{step}\} \quad (18)$$

wobei das Hadamard-Produkt ( $\odot$ ) zur elementweisen Multiplikation verwendet wird<sup>25</sup>.

Danach wird basierend auf dem initialen Lösungspunkt  $x = x_{start}$  die Starttemperatur für jede Zielfunktion wie folgt bestimmt:

$$\forall i \in \{1, \dots, z\}: c_{i,0} \cong \frac{\Delta_{f_i}^{(+)}}{\ln\left(\frac{m_2}{m_2\chi - m_1(1-\chi)}\right)} \quad (19)$$

Dabei ist die Anfangsakzeptanz zu  $\chi = 0.8$  gewählt um ein hohes Maß an Erkundung zu erlauben. Zudem werden zehn Punkte  $x_{umg} \in \Omega_{umg}(x_{start}) \subset \Omega$  aus der Umgebung von  $x_{start}$  durch Ausführung der Simulation ausgewertet. Dabei wird in  $m_1$  hinterlegt in wie vielen dieser Punkte sich die Zielfunktion verbessert und in  $m_2$  wie oft sich diese verschlechtert. Außerdem wird in  $\Delta_{f_i}^{(+)}$  die durchschnittliche Abweichung zwischen  $f_i(x_{umg})$  und  $f_i(x_{start})$  festgehalten.

Anschließend wird ein neuer Punkt  $y$  aus der Umgebung von  $x$  bestimmt, so kann die Lösung potenziell verbessert werden, sofern  $x$  nicht bereits zur Pareto-Front gehört. Für die Auswahl eines Punktes aus dieser Umgebung erfolgt die Einführung eines Zufalls- bzw. Richtungsvektor  $r = (r_1, \dots, r_n)^T \in \{-1, 0, 1\}^n$  für den die folgende Wahrscheinlichkeitsfunktion gilt:

$$\forall a \in \{-1, 0, 1\} \forall i \in \{1, \dots, n\}: P(r_i = a) = \frac{1}{3} \quad (20)$$

Demensprechend kann ein neuer Punkt  $y \in \Omega_{umg}(x)$  mit Hilfe der folgenden Formel bestimmt werden:

$$y = x + r \odot x_{step} \quad (21)$$

Somit besteht für jeden Punkt in der Umgebung von  $x$  dieselbe Wahrscheinlichkeit als neuer Punkt ausgewählt zu werden. Anschließend erfolgt die Ausführung der Simulation mit den Werten des Punktes  $y$  für die Produktionsparameter um die Produktionsaufwände  $F(y)$  zu ermitteln. Dabei wird eine Liste aller Punkte  $y$  und den zugehörigen Produktionsaufwänden  $F(y)$  geführt und in jeder Iteration des MOSA entsprechend erweitert. Diese Liste ermöglicht es zum einen,

---

<sup>25</sup> Eine zweidimensionale Visualisierung der Umgebung eines Punktes ist im Anhang in Abbildung B.6 aufgeführt.

die doppelte Auswahl und Betrachtung eines Punktes zu vermeiden. Zum anderen kann sie verwendet werden, um im Anschluss an das MOSA sowohl die approximierte Pareto-Front  $\tilde{P}_f$  als auch die zugehörige Pareto-Menge  $\tilde{P}$  zu ermitteln. Zudem kann basierend auf der Ermittlung von  $F(y)$  geprüft werden, ob sich mindestens ein Zielkriterium (Zeit, Kosten, Energie) verbessert hat. Ist dies der Fall, wird der Richtungsvektor  $r$ , mit dem Ziel einer Annäherung an die Pareto-Front, nach Möglichkeit beibehalten. Für die Entscheidung ob ein betrachteter Punkt  $y$  als neuer Lösungspunkt gewählt wird (also  $x = y$ ), wird zunächst geprüft ob der Punkt  $y$  den Punkt  $x$  dominiert, ist dies der Fall so wird  $y$  übernommen. Ist dies nicht der Fall, so wird  $y$  basierend auf [157] mit der folgenden Akzeptanzwahrscheinlichkeit übernommen:

$$P(y) = \alpha \prod_{i=1}^z \min \left\{ 1, e^{-w_i \frac{f_i(y) - f_i(x)}{c_i}} \right\} + (1 - \alpha) \min \left\{ 1, \max_{i \in \{1, \dots, z\}} \left\{ e^{-w_i \frac{f_i(y) - f_i(x)}{c_i}} \right\} \right\} \quad (22)$$

Mithilfe von  $\alpha \in [0,1]$  kann die Gewichtung des ersten bzw. zweiten Summanden eingestellt werden. Ein hoher Wert von  $\alpha$  gibt dem ersten Summanden mehr Gewicht wodurch bessere Lösungen bevorzugt akzeptiert werden, dies entspricht einer Intensivierung (Exploitation). Ein niedriger Wert von  $\alpha$  hingegen gibt dem zweiten Summanden mehr Gewicht wodurch auch schlechtere Lösungen öfter akzeptiert werden, dies entspricht einer erhöhten Diversität (Exploration)<sup>26</sup>. Für die zur Evaluierung herangezogene Realisierung wurde  $\alpha = 0.5$  gewählt. Für den Temperaturwert  $c = (c_1, \dots, c_z)$  gilt, dass für höhere Temperaturen die Akzeptanzwahrscheinlichkeit größer ist als für geringe. Da die Temperatur in jeder Iteration reduziert wird, wird somit in frühen Iterationen eine explorative Suche begünstigt, wohingegen in späten Iterationen die Intensität erhöht wird<sup>27</sup>. Über die Gewichtung  $w_i$  kann der Einfluss der Veränderung eines Zielkriteriums auf die Akzeptanzwahrscheinlichkeit  $P(y)$  eingestellt werden, je höher die Gewichtung gewählt wird, desto größer ist der Einfluss des jeweiligen Kriteriums. Die Anzahl der Punkte welche in dieser beschriebenen inneren Schleife des MOSA betrachtet werden ist über die Länge der Markov-Kette  $L_m$  festgelegt welche in der zur Evaluierung genutzten Realisierung für jede Iteration als eins gewählt wurde. Anschließend erfolgt die Reduktion der Temperatur, wobei jede Zielfunktion basierend auf der individuellen initialen Temperatur einen individuellen Abkühlungsprozess erfährt. Hierzu gilt für die  $i$ -te Zielfunktion in der  $m$ -ten Iteration:

$$c_{i,m} = c_{i,0} \beta^m \quad (23)$$

---

<sup>26</sup> Der Einfluss von  $\alpha$  auf die Akzeptanzwahrscheinlichkeit  $P(y)$  ist im Anhang in Abbildung B.7 exemplarisch visualisiert.

<sup>27</sup> Der Einfluss der Temperatur  $c$  auf die Akzeptanzwahrscheinlichkeit  $P(y)$  ist im Anhang in Abbildung B.8 exemplarisch visualisiert.

Der Abkühlparameter  $\beta$  wird dabei über folgende Formel bestimmt:

$$\beta = \left( \frac{c_{i,m_{max}}}{c_{i,0}} \right)^{\frac{L_m}{m_{max}}} \quad (24)$$

Dabei ergibt sich  $m_{max}$  aus der über die RM-GUI (siehe Abbildung 4.1 – (6)) eingestellten Anzahl an Iterationen und die Endtemperatur wurde mit  $c_{i,end} = c_{i,m_{max}} = \frac{1}{1000} c_{i,0}$  als ein Tausendstel der Starttemperatur gewählt. Der MOSA-Algorithmus wird beendet sofern nach der Temperaturreduktion einer Iteration die Endtemperatur erreicht wurde ( $c_{i,m} \cong c_{i,end}$ ), andernfalls fährt der MOSA-Algorithmus in der nächsten Iteration mit der Bestimmung eines neuen Punktes  $y$  entsprechend des beschriebenen Schemas fort. Abschließend erfolgt wie oben bereits angedeutet, die Bildung der approximierten Pareto-Front  $\tilde{P}_f$  und der zugehörigen Pareto-Menge  $\tilde{P}$  indem (mit Hilfe des MATLAB Skripts *findParetoOptimal.m*) die nicht-dominierten Produktionsaufwände  $F(y)$  sowie die zugehörigen Punkte  $y$  aus der geführten Liste bestimmt werden.

Wurde bei der Auswahl des Optimierungsverfahrens über die RM-GUI (siehe Abbildung 4.1 – (5)) ein Verfahren gewählt, welches wie z.B. das MOSA oder der multikriteriellen Genetischen Algorithmus eine Pareto-Front  $\tilde{P}_f$  approximiert, so muss anschließend eine Kompromisslösung  $\tilde{x}^*$  aus der Pareto-Menge  $\tilde{P}$  ausgewählt werden. Dies geschieht unter Verwendung der Gewichtung  $w_i$  der Zielkriterien (Zeit, Kosten und Energie) die über die RM-GUI (siehe Abbildung 4.1 – (4)) eingestellt werden kann. Zudem kann über die RM-GUI (siehe Abbildung 4.1 – (7)) eine von drei Optionen für die Auswahl von  $\tilde{x}^*$  gewählt werden. Die erste Option (Radiobutton *weighed sum*) ist die Auswahl basierend auf der Methode der gewichteten Summe<sup>28</sup>, wobei  $\tilde{x}^*$  wie folgt aus der Pareto-Menge ermittelt wird:

$$\tilde{x}^* = \min_{x \in \tilde{P}} \sum_{i=1}^z w_i f_{i,norm}(x) \quad (25)$$

Um die Normierung der Zielfunktion  $f_{i,norm}(x)$  zu erhalten, werden die während der Optimierung erhaltenen Maximal- und Minimalwerte der Zielfunktion wie folgt herangezogen:

$$\forall i \in \{1, \dots, z\}: f_{i,norm}(x) = \frac{f_i(x) - \min_{x \in \tilde{P}} f_i(x)}{\max_{x \in \tilde{P}} f_i(x) - \min_{x \in \tilde{P}} f_i(x)} \quad (26)$$

Ebenso wie für den Einsatz der Methode der gewichteten Summe für die Lösung eines multikriteriellen Optimierungsproblems mit Hilfe eines einkriteriellen Ersatzproblems, gilt auch hier das

---

<sup>28</sup> Auch bekannt als Simple Additive Weighting (SAW).

eine Kompromisslösung  $\widetilde{x}^*$  nur dann zuverlässig gefunden werden kann, wenn es sich um eine konvexe Pareto-Front handelt<sup>29</sup>. Daher basiert die zweite Option (Radiobutton *L\_infnorm*) für die Auswahl von  $\widetilde{x}^*$  auf der Methode der gewichteten Metrik, für  $p = \infty$ , wie im Fall dieser Realisierung, ist diese Methode auch als *Compromise Ranking Method* bekannt [158]. Da mit Hilfe der  $L_\infty$ -Norm für konvexe und auch konkave Pareto-Fronten Kompromisslösungen gefunden werden können, werden hier in Anlehnung an das Tchebycheff-Problem die Kompromisslösungen  $\widetilde{x}^*$  folgendermaßen bestimmt:

$$\widetilde{x}^* = \min_{x \in P} \max_{i \in \{1, \dots, z\}} \{w_i(f_{i,norm}(x) - 0)\} \quad (27)$$

Auch hier gilt die Normierung der Zielfunktion  $f_{i,norm}(x)$  die bereits für die erste Option beschrieben wurde. Es ist jedoch festzuhalten das selbst für konvexe Pareto-Fronten Kompromisslösungen gewählt werden, welche bei gleichbleibender Gewichtung von der Wahl basierend auf der ersten Option abweichen<sup>30</sup>. Da das Auswahlverhalten der ersten Option besser mit der in Kapitel 3.2.3 beschriebenen Nutzwertanalyse zur späteren Bewertung und Auswahl von alternativen Systemkonfigurationen harmoniert, wird diese präferiert. Daher wurde eine dritte Option (Radiobutton *weighted sum and L\_infnorm*) realisiert, welche die beiden anderen Optionen kombiniert. Dabei wird die erste Option verwendet sofern eine konvexe Pareto-Front vorliegt und andernfalls die zweite Option. Um zu ermitteln ob eine betrachtete Pareto-Front aus einer Mischform aus konvexen und konkaven Anteilen besteht wird eine konvexe Hülle um die Punktmenge der Pareto-Front gelegt<sup>31</sup>. Diese konvexe Hülle ist dabei definiert, als die kleinste konvexe Menge welche die Punktmenge der Pareto-Front enthält. Sofern alle Punkte der Pareto-Front auf dem Rand der konvexen Hülle liegen, so kann eine Mischform ausgeschlossen werden. Liegt eine Mischform vor, so wird die zweite Option (gewichtete Metrik) zur Ermittlung der die Kompromisslösung  $\widetilde{x}^*$  eingesetzt. Liegt keine Mischform vor, kann die Pareto-Front konvex oder konkav sein, wobei, wie oben beschrieben, eine konkave Pareto-Front für die Anwendung der ersten Option (gewichtete Summe) problematisch ist. Daher wird die Kompromisslösung  $\widetilde{x}^*$  zunächst durch die erste Option ermittelt und anschließend geprüft ob es sich um eine Extremlösung ( $\forall i: f_i(\widetilde{x}^*) = \min_{x \in P} f_i(x) \vee f_i(\widetilde{x}^*) = \max_{x \in P} f_i(x)$ ) handelt. Ist dies der Fall, so wird die Kompromisslösung  $\widetilde{x}^*$  stattdessen erneut, dieses Mal unter Verwendung der zweiten Option, ermittelt und diese Lösung verwendet. Der Ablauf für die Auswahl basierend auf der dritten Option ist in Abbildung 4.14 zusammengefasst.

---

<sup>29</sup> Eine Veranschaulichung dieser Problematik ist im Anhang in Abbildung B.10 abgebildet.

<sup>30</sup> Um eine Veranschaulichung dieser Gegebenheit zu bieten ist im Anhang die Abbildung B.11 gegeben.

<sup>31</sup> Zwei Beispiele für eine konvexe Hülle um die Punktmenge einer Pareto-Front gibt Abbildung B.12 im Anhang.

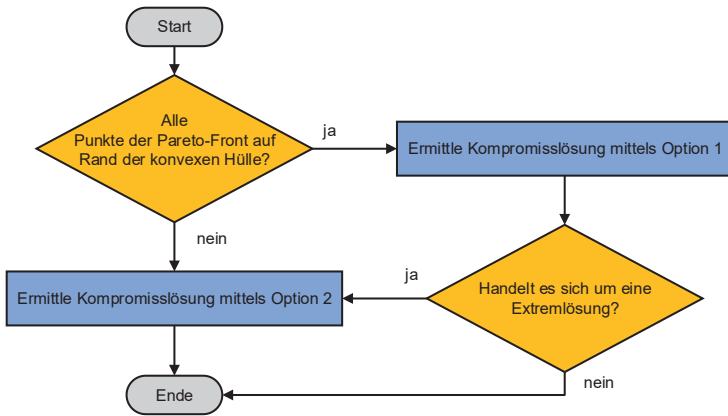


Abbildung 4.14: Auswahl der Kompromisslösung  $\tilde{x}^*$  basierend auf der dritten Option

An dieser Stelle sei angemerkt, dass die durch das Optimierungsverfahren ermittelte Lösung  $\tilde{x}^*$  dem oben als  $x^*$  bezeichneten Satz der optimierten Produktionsparametern entspricht ( $x^* = \tilde{x}^*$ ) und für die zugehörigen Produktionsaufwände bzw. das Optimierungsergebnis  $F(x^*) = F(\tilde{x}^*)$  gilt.

Der **Evaluation Agent**, dessen Ablauf in Abbildung 4.15 abgebildet ist, realisiert die in Kapitel 3.2.3 und Kapitel 3.2.4 beschriebene Bewertung und Auswahl von Konfigurationen und wird bereits bei der Initialisierung des Agentensystems gestartet. Danach wartet er auf eine Nachricht vom SBO Agent welche wie beschrieben eine optimierte Systemkonfiguration und alle zugehörigen Informationen beinhaltet. Auch hier kommt wieder eine Message Queue in Kombination mit einem *CyclicBehaviour* zum Einsatz, sodass der Evaluation Agent die subsequenten Schritte durchführen kann, ohne dass eine vom SBO Agent gesendete Nachricht, also eine ermittelte Systemkonfiguration, verloren geht. Anschließend erfolgt die in Kapitel 3.2.3 beschriebene Nutzwertanalyse mit Hilfe eines *OneShotBehaviours* indem zunächst der kriterienspezifische Gesamtaufwand  $a_{z,tot}$  der betrachteten Systemkonfiguration durch eine Addition ihrer kriterienspezifischen Rekonfigurations-  $a_{z,reconfiguration}$  und Produktionsaufwände  $a_{z,production} = f_z(x^*)$  jeweils für Zeit, Kosten und Energie bestimmt wird. Danach wird die betrachtete Systemkonfiguration einer Liste (*LinkedList*) hinzugefügt die potenziell bereits bewertete Systemkonfigurationen umfasst (bei der Betrachtung der ersten optimierten Systemkonfiguration ist diese Liste leer). Im Anschluss, erfolgt für jedes Kriterium eine Iteration über alle Systemkonfigurationen um den höchsten kriterienspezifischen Gesamtaufwand  $a_{z,tot,max}$  und den niedrigsten kriterienspezifischen Gesamtaufwand  $a_{z,tot,min}$  zu ermitteln. Anschließend erfolgt die Zuweisung eines normierten Aufwandswerts  $r_z$  für alle Systemkonfigurationen und jeden kriterienspezifischen Gesamtaufwand  $a_{z,tot}$  nach der in Kapitel 3.2.3 angeführten Formel auf den, durch die ermittelten höchsten

( $r_z = 0$ ) und niedrigsten ( $r_z = 1$ ) Werte für den kriterienspezifischen Gesamtaufwand, definierten Wertebereich. Dabei werden für jede Systemkonfiguration die drei Aufwandswerte  $r_z$  (für Zeit, Kosten und Energie) in einer *LinkedList* gespeichert. Danach wird der Nutzwert  $v$  für jede Systemkonfiguration wie in Kapitel 3.2.3 beschrieben mit Hilfe der Formel  $v = \sum_z w_z * r_z$  berechnet, wobei  $w_z$  die Kriteriengewichtung die in der RM-GUI (siehe Abbildung 4.1 – (4)) eingestellt wurde beinhaltet und  $\sum_z w_z = 1$  gilt. Basierend auf den so ermittelten Nutzwerten aller Systemkonfigurationen wird die erwähnte Liste der betrachteten Systemkonfigurationen so sortiert, dass das erste Element die Systemkonfiguration mit dem höchsten Nutzwert ist. So wird die beste Systemkonfiguration identifiziert und anschließend in Form einer XML-Datei ausgegeben. Das beschriebene Verfahren hat den Vorteil, dass, sobald mindestens eine Systemkonfiguration durch den Evaluation Agent bewertet wurde, zu jeder Zeit die beste bekannte Systemkonfiguration vorliegt und angewendet werden kann, selbst wenn das Rekonfigurationsmanagement vorzeitig beendet werden sollte. Die ausgegebene XML-Datei beinhaltet dabei insbesondere Informationen über die Position und Konfiguration aller eingesetzten CPPMs, sowie die Fertigungsservices inklusive der benötigten Parametrierung (dies umfasst sowohl zuvor optimierte als auch fixe Parameter) und Reihenfolge der Services im Sinne einer BOP. Zudem werden die Gesamtaufwände (in Zeit, Kosten und Energie) sowie der Nutzwert und das geforderte Auftragsvolumen ausgegeben. Zuletzt prüft auch der Evaluation Agent ob sich bereits weitere (optimierte) Systemkonfigurationen in der angesprochenen Message Queue befinden, ist dies der Fall führt er wiederum das beschriebene Prozedere für die nächste Systemkonfiguration durch. Ist jedoch keine weitere Systemkonfiguration vorhanden, so wartet der Evaluation Agent auf den Erhalt einer weiteren Nachricht des SBO Agent.

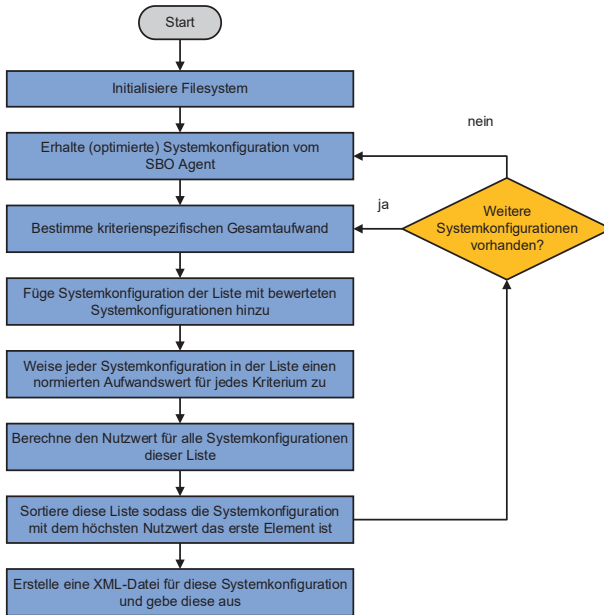


Abbildung 4.15: Programmablaufplan des Evaluation Agent

Eine detaillierte Beschreibung der vom Evaluation Agent ausgegebenen XML-Datei wird in Kapitel 5.3 anhand des Beispiels in Abbildung 5.5 gegeben, das sich aus den für die Evaluation eingesetzten Produktionsmodulen und dem betrachteten Produktionsauftrag ergibt.

## 4.2 Realisierung des Proxy-Layers

Was das Proxy-Layer anbelangt, so wird beim Systemstart für jede verfügbare XML-Datei, in der sich jeweils die Informationen eines der vorhandenen CPMs befinden, die *Simple API for XML (SAX)* verwendet, um die Informationen der XML-Dateien zu parsen und in das System zu integrieren (siehe Schritt 5 in Abbildung 4.16). Dadurch stehen die Modelle zur Verfügung und können wie in Kapitel 3.2 und Kapitel 4.1 beschrieben verwendet werden.

Um sicherzustellen, dass die XML-Dateien dem in Kapitel 3.3 beschriebenen einheitlichen CPM-Informationsmodell entsprechen, und um Tool-Support für deren aufwandsreduzierte Erstellung zur Verfügung zu stellen, wird das *Eclipse Modeling Framework (EMF)* verwendet.





### 4.3 Realisierung des Control- und des Asset-Layers

Die Realisierung des Control- und des Asset-Layers ist von dem in [159] vorgestellten physischen modularen Produktionssystem und dessen SOA-basierter Steuerung inspiriert. Sie verwendet das *Proslys OPC UA SDK for Java* für die Umsetzung der SBC durch ein OPC UA Steuerungsnetzwerk. Die prototypische Implementierung umfasst ein modulares Produktionssystem mit einem Matrix-Layout, das in Unity simuliert wird (siehe Abbildung 4.17). Das Produktionssystem bietet diskrete Fertigungsservices wie beispielsweise Bohren, Fräsen oder Stanzen und weist eine hohe Rekonfigurierbarkeit und Flexibilität auf.

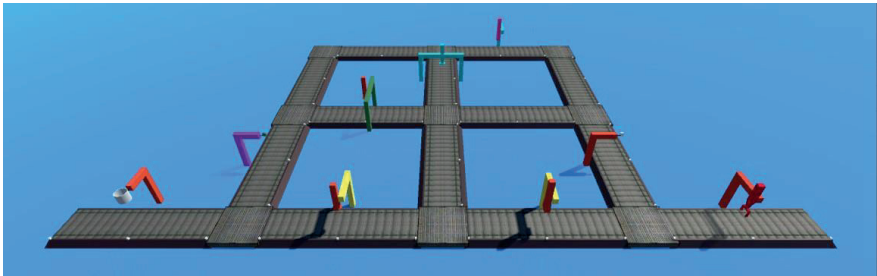


Abbildung 4.17: Unity-Simulation des modularen Produktionssystems [75]

Eine Übersicht über das Funktionsprinzip der SBC ist in Abbildung 4.18 skizziert. Da die Module (der CPPMs und der CPTMs) als Service Provider fungieren, werden sie sowohl unter Verwendung einer OPC UA Client- als auch einer OPC UA Server-Komponente realisiert. Die Client-Komponente kümmert sich um die De-/Registrierung des Moduls und seiner Services beim Discovery Server. Die Server-Komponente bietet hingegen die Services, d. h. Fertigungs- oder Transportservices, eines Moduls an, die von den intelligenten Produkten in Anspruch genommen werden können. Der Discovery-Server stellt einen Service-Broker dar und nutzt eine MariaDB-Datenbank zur Datenspeicherung und eine OPC-UA-Server-Komponente zur dynamischen Bearbeitung der Anfragen.

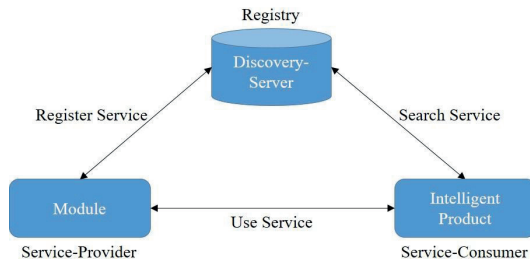


Abbildung 4.18: Funktionsprinzip der eingesetzten SBC [75]

Die Abarbeitung eines konkreten Produktionsauftrages in Form einer BOP durch das OPC UA Steuerungsnetzwerk entsprechend der Steuerungslogik der SBC ist in vereinfachter Form in Abbildung 4.19 durch ein Sequenzdiagramm beschrieben. Die intelligenten Produkte werden, wie in Kapitel 3.3 beschrieben, entsprechend des Produktionsauftrags und auf der Grundlage des vom Evaluation Agent gelieferten Ergebnisses instanziiert, das die abgeleitete BOP mit den optimierten Produktionsparametern beinhaltet. Ein weiteres intelligentes Produkt wird dabei immer erst dann instanziiert und über das Source-Modul in das Produktionssystem eingebracht, wenn das vorherige Produkt das erste Transportmodul in der Unity-Simulation wieder verlassen hat. Darauf aufbauend suchen die OPC UA Clients der einzelnen intelligenten Produkte mit Hilfe des Discovery Servers geeignete Produktionsmodule die die Fertigungsservices entsprechend ihrer BOP anbieten. Danach wird der Belegstatus aller geeigneten Produktionsmodule abgefragt. Anschließend erfolgt die Wegfindung basierend auf dem Dijkstra-Algorithmus und der kürzeste Weg in Abhängigkeit der verfügbaren Transportmodule (vereinfachend ist lediglich ein Transportmodul abgebildet) wird gewählt. Subsequent werden die gewählten Transportmodule und das gewählte Produktionsmodul reserviert. Dann werden nacheinander die benötigten Transportservices und zuletzt der benötigte Fertigungsservice aufgerufen. Der Ablauf dieses Schrittes wiederholt sich bis die BOP vollständig abgearbeitet ist. Abschließend erfolgt der Abtransport zum Sink-Modul und das Entfernen des Produktes aus dem Produktionssystem. Die Interaktion mit dem Source- und dem Sink-Modul, sowie die Freigabe der Module nach erfolgreicher Ausführung ihrer Services wurde zugunsten einer besseren Übersichtlichkeit nicht abgebildet.

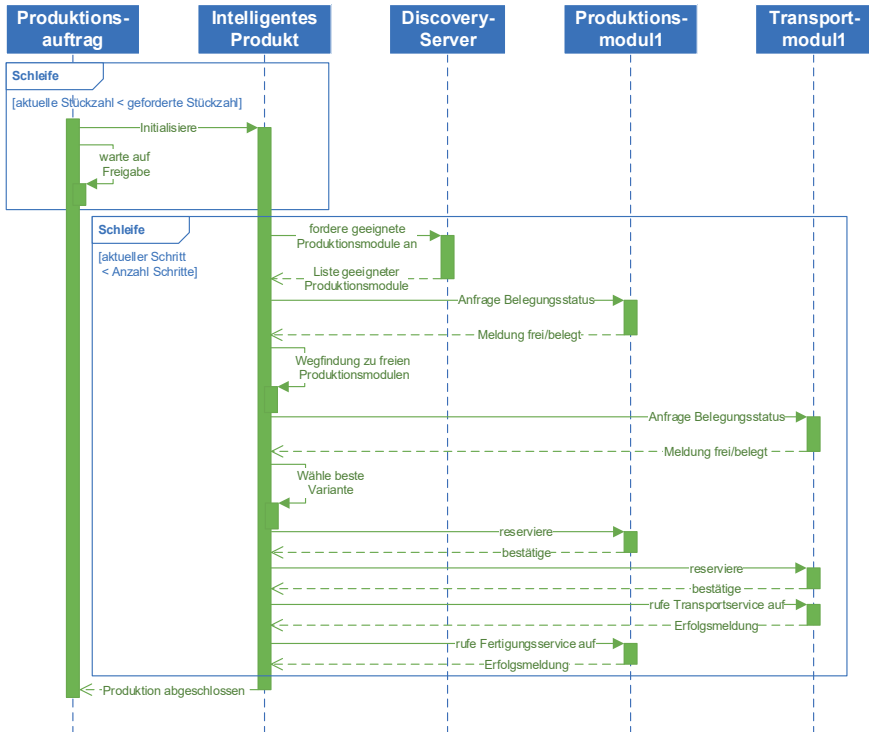


Abbildung 4.19: Abarbeitung eines Produktionsauftrags durch die SBC (vereinfacht)

Die Ausführung der Services erfolgt über eine Socket-Kommunikation mittels TCP/IP-Verbindung zwischen den Modulen des Steuerungsnetzwerks und ihrem Gegenstück in der Unity-Simulation, wobei C# zur Realisierung des Produktionsablaufs innerhalb der Simulation verwendet wird.

Zudem wurde ein Monitoring zur Messung und Darstellung der Rekonfigurations- und Produktionsaufwände implementiert, welches für die Evaluierung eingesetzt wurde und dessen Einsatz daher in Kapitel 5.3 näher aufgezeigt wird<sup>32</sup>. Für die Berechnung der Rekonfigurationsaufwände kann über die Buttons *Start Reconfiguration* und *End Reconfiguration* das Tracking von Änderungen an der Unity-Simulation gestartet bzw. beendet werden. Dabei werden Rekonfigurations-

<sup>32</sup> Zudem sind weiterführende Abbildungen und Informationen über die Unity-Simulation und das Monitoring im Anhang aufgeführt.

aufwände auf der Maschinen- und Systemebene basierend auf den Änderungen am Produktionssystem durch den Benutzer berechnet. Hierzu kann der Benutzer sowohl vorhandene Module Entfernen (löschen) als auch Module in ihrer gewünschten Modulkonfiguration über ein Dropdown Menu per Drag & Drop hinzufügen. Die Berechnung der Produktionsaufwände sowie deren Anzeige parallel zur Simulation der Produktion erfolgt ebenso in der Unity-Simulation und wird über das Steuerungsnetzwerk veranlasst. Basierend auf den Prozessdaten können sowohl Zeit, Kosten als auch Energie zur Laufzeit dargestellt werden.

Für die Interaktion mit dem Steuerungsnetzwerk wurde ein Webinterface mittels HTML, CSS und PHP realisiert (s. Abbildung 4.20). Zudem wurde XAMPP verwendet, um der Einrichtung und Verwaltung des Apache HTTP Webservers und des DBMS (MySQL) zu vereinfachen. Das Webinterface ermöglicht die insbesondere im Zuge der Evaluation erforderliche Eingabe konkreter Produktionsaufträge im Sinne einer parametrisierten BOP, entsprechend dem Produktionsauftrag und auf Basis des vom Evaluation Agent gelieferten Ergebnisses. Dabei können die Fertigungsservices der Produktionsmodule genutzt werden, die sich aktuell im Produktionssystem (Unity-Simulation) befinden. Hierzu werden die Daten des Discovery-Servers genutzt welche in der MySQL-Datenbank des DBMS vorgehalten werden. Zudem wird auch eine Überwachung des Produktionssystems, seiner Module und deren Services sowie des Status aktiver Produktionsaufträge über das Webinterface ermöglicht.

The screenshot displays the 'Webinterface of OPC-UA-Controlnetwork'. It features a sidebar with 'Produktionssystem' and 'Productionorders' buttons. The main area is titled 'Detail view of a productionorder' and includes fields for Name, Quantity (projected/finished), Product, and an 'Active' checkbox. Below these are buttons for 'Create new Product', 'Edit Product', and 'Delete Product'. A section for 'Available services' shows a dropdown menu with 'ServiceMetalllogoStanzen' and a 'Select Service' button. A 'Projected production steps' table is also visible. Two callout boxes highlight the 'Übersicht der BOP' (Overview of the BOP) and the 'Auswahl der Fertigungs-services' (Selection of manufacturing services).

**Webinterface of OPC-UA-Controlnetwork**

Active productionorders | Saved productionorders | New productionorder

**Detail view of a productionorder**

Name:  Quantity (projected):

Quantity (finished):

Product:

Available services

Projected production steps:

| Step | Service                      | Parameter      |     |
|------|------------------------------|----------------|-----|
| 0    | ServiceKunststoffBohrenA     | Set Parameters | Del |
| 1    | ServiceKunststofflogoStanzen | Set Parameters | Del |

**Übersicht der BOP**

**Auswahl der Fertigungs-services**

Abbildung 4.20: Webinterface der SBC - Produktionsauftragsübersicht

So stellt die Realisierung des Asset- und des Control-Layers zum einen ein eigenständiges, dezentral gesteuertes, modulares Produktionssystem dar, das über das Webinterface bedient werden kann<sup>33</sup>. Zum anderen stellt sie eine (von mehreren möglichen) Umsetzung der notwendigen Basis für die Realisierung des gesamten CPPS mit der in Kapitel 3.4 vorgestellten Architektur dar.

## 4.4 Systemimplementierung

In diesem Unterkapitel wird eine detaillierte Beschreibung der eingesetzten Hard- und Software angegeben.

Die beschriebene Realisierung wurde auf einem Personal Computer mit einer Intel Core i7-7700 CPU @ 3.60 GHz (8 CPUs) und 65536MB RAM Speicher implementiert und evaluiert. Das eingesetzte Betriebssystem ist Windows 10. Die Eclipse IDE v4.16.0 wurde für die Entwicklung des MAS und der SBC verwendet. Die IDE dient außerdem zur Durchführung des oben beschriebenen Workflows zur Modellierung und Erstellung von XML-Dateien, weshalb das Eclipse Modeling Framework SDK v2.5.0 installiert wurde. Hinsichtlich Java kommen das JDK v1.8.0\_291 sowie das JRE v1.8.0\_311 für das MAS, EMF und die SBC zum Einsatz. Die Version des erwähnten Prosys OPC UA SDK for Java ist v2.2.2-638, die Datenbank des Discovery-Servers der SBC ist MariaDB v10.1.37. Bezüglich MATLAB wurde die Version 2021b mit Simulink v10.1, Stateflow v10.2 und der Optimization Toolbox v8.5 gewählt. Die Unity-Version für die Simulation des modularen Produktionssystems ist v2018.2.0f2. Die in Unity genutzten 3D-Objekte wurden mit Blender erstellt.

---

<sup>33</sup> Eine detaillierte Übersicht der Umsetzung des Control- und des Asset-Layers mit Anbindung des zugehörigen Webinterfaces ist im Anhang in Abbildung B.15 dargestellt.

## 5 Evaluierung des Konzepts

In diesem Kapitel wird die durchgeführte Evaluierung des Konzepts beschrieben. Dazu wird in Kapitel 5.1 zunächst der Ablauf der Evaluierung im Rahmen des Design Zyklus skizziert und in Kapitel 5.2 bis Kapitel 5.4 ausgeführt. In Kapitel 5.5 erfolgt abschließend ein *Abgleich mit den definierten Anforderungen an die Konzeption*, aber auch eine Beurteilung, ob die formulierte Forschungsfrage hinreichend beantwortet wurde, welche dem Relevanz Zyklus zuzuordnen ist.

Der EMF-basierte Workflow zur Modellierung und Erstellung von XML-Dateien, der in Kapitel 4 dargestellt und beschrieben wird, wurde zur Erstellung der XML-Dateien für die einzelnen CPMs eingesetzt. Im gegenwärtigen Zustand, welcher die Basis der Evaluierung darstellt, sind für das prototypisch implementierte modulare Produktionssystem sieben rekonfigurierbare CPPMs<sup>34</sup> mit 20 Konfigurationsalternativen verfügbar, die 64 Services anbieten. Unter Berücksichtigung des vorgegebenen Layouts und abhängig vom jeweiligen Produktionsauftrag werden bis zu 3,2 Millionen mögliche Lösungen verwaltet.

### 5.1 Ablauf der Evaluierung

Die durchgeführte Evaluierung gliedert sich in die drei Schritte *Definition von Evaluationsszenarien*, *Durchführung der Evaluierungsszenarien* und *Auswertung der Ergebnisse der Evaluierungsszenarien*. Eine Übersicht dieser drei Schritte, welche zugleich die Struktur der nachfolgenden drei Teilkapitel vorgibt, ist in Abbildung 5.1 gegeben.

---

<sup>34</sup> Beispiele für Produktionsmodule sind etwa Bohren, Fräsen oder Stanzen. Exemplarisch wird das Bohrmodul, insbesondere im Hinblick auf die Bestimmung der Aufwände, im Anhang ausführlicher beschrieben.

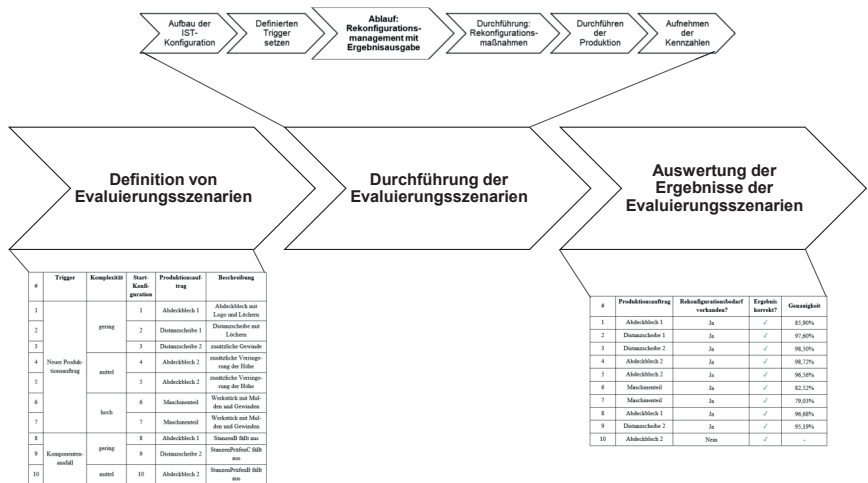


Abbildung 5.1: Ablauf der Evaluierung

## 5.2 Definition von Evaluierungsszenarien

Die Definition der Evaluierungsszenarien umfasst zehn Szenarien (siehe Tabelle 5.1), von denen jedes durch den gewählten *Trigger*, die implizite *Komplexität*, die verwendete *Start-Konfiguration*, den gewählten *Produktionsauftrag* und eine kurze *Beschreibung* spezifiziert wird.



**Tabelle 5.1: Überblick über die definierten Evaluierungsszenarien**

| #  | Trigger                  | Komplexität | Start-Konfiguration | Produktionsauftrag | Beschreibung                      |
|----|--------------------------|-------------|---------------------|--------------------|-----------------------------------|
| 1  | Neuer Produktionsauftrag | gering      | 1                   | Abdeckblech 1      | Abdeckblech mit Logo und Löchern  |
| 2  |                          |             | 2                   | Distanzscheibe 1   | Distanzscheibe mit Löchern        |
| 3  |                          |             | 3                   | Distanzscheibe 2   | zusätzliche Gewinde               |
| 4  |                          | mittel      | 4                   | Abdeckblech 2      | zusätzliche Verringerung der Höhe |
| 5  |                          |             | 5                   | Abdeckblech 2      | zusätzliche Verringerung der Höhe |
| 6  |                          | hoch        | 6                   | Maschinenteil      | Werkstück mit Mulden und Gewinden |
| 7  |                          |             | 7                   | Maschinenteil      | Werkstück mit Mulden und Gewinden |
| 8  | Komponentenausfall       | gering      | 8                   | Abdeckblech 1      | StanzenB fällt aus                |
| 9  |                          |             | 9                   | Distanzscheibe 2   | StanzenPrüfenC fällt aus          |
| 10 |                          | mittel      | 10                  | Abdeckblech 2      | StanzenPrüfenB fällt aus          |

Die definierten Evaluierungsszenarien decken ein relativ breites Spektrum ab. Beide Trigger-Arten, d.h. Anforderungsänderungen (durch neue Produktionsaufträge) und Komponentenausfälle, wurden mehrfach evaluiert. Es wurden unterschiedlich komplexe Szenarien untersucht, wobei die Komplexität durch die Anzahl der betrachteten alternativen Systemkonfigurationen, die optimiert werden, definiert wird. Dabei steht 0 bis 10.000 für eine geringe, 10.001 bis 50.000 für eine mittlere und >50.000 für eine hohe Komplexität. Da sich die Systemkonfigurationen des CPPS in der Realität ständig ändern, wurden verschiedene Start-Konfigurationen als Ausgangspunkt betrachtet (siehe auch Evaluierungsszenarien 6 und 7). Daraus ergeben sich unterschiedliche Rekonfigurationsaufwände, die dazu führen, dass verschiedene Alternativen als Gewinner hervorgehen und somit zum Einsatz kommen. Darüber hinaus wurden verschiedene Produktionsaufträge verwendet, die z.B. die Produktion von Maschinenteilen, Distanzscheiben oder Abdeckblechen, auch in mehreren Ausprägungen, umfassen.

## 5.3 Durchführung der Evaluierungsszenarien

Die **Durchführung der Evaluierungsszenarien** erfolgt nach dem in Abbildung 5.2 dargestellten dem Muster in sechs Schritten:

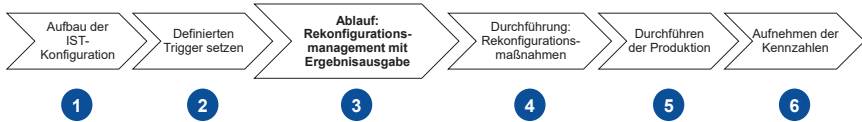


Abbildung 5.2: Durchführung der Evaluierungsszenarien

- 1) Aufbau der definierten IST-Konfiguration (Startkonfiguration): Die definierte Startkonfiguration (zwei Beispiele sind in Abbildung 5.3 dargestellt) wird auf das Produktionssystem in Unity übertragen. Dazu kommt die in Kapitel 4.3 beschriebene Drag & Drop-Funktionalität der Unity-Simulation zum Einsatz<sup>35</sup>. Die Initialisierung des CPPS umfasst die automatische Integration der XML-Dateien, sowie die automatische Erfassung der aktuell eingesetzten Systemkonfiguration.

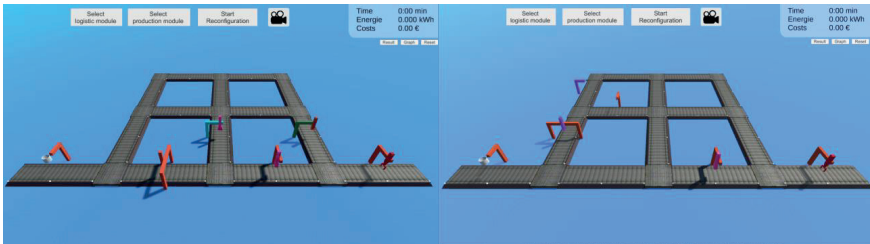


Abbildung 5.3: Beispiele für unterschiedliche Startkonfigurationen des modularen Produktionssystems

- 2) Setzen des definierten Triggers: Der Auslöser des jeweiligen Szenarios, also ein Komponentenausfall oder ein neuer Produktionsauftrag wird eingespeist. Hierzu ist in Abbildung 5.4 eine Übersicht von vier der fünf Produktionsaufträge *Abdeckblech 1*, *Abdeckblech 2*, *Distanzscheibe 1*, *Distanzscheibe 2* und *Maschinenteil* die in der Evaluierung eingesetzt wurden gegeben. Die detaillierte Beschreibung eines Produktionsauftrages in Form der merkmalsbasierten Eingangs- und Ausgangsproduktbeschreibung sowie die zugehörige XML-Datei sind aufgrund ihres Umfangs in Abbildung C.1 des Anhangs am Beispiel des Produktionsauftrags *Distanzscheibe 1* dargestellt.

<sup>35</sup> Abbildung B.17 des Anhangs zeigt eine Detailansicht der Unity-GUI zur Auswahl der Modulkonfigurationen.

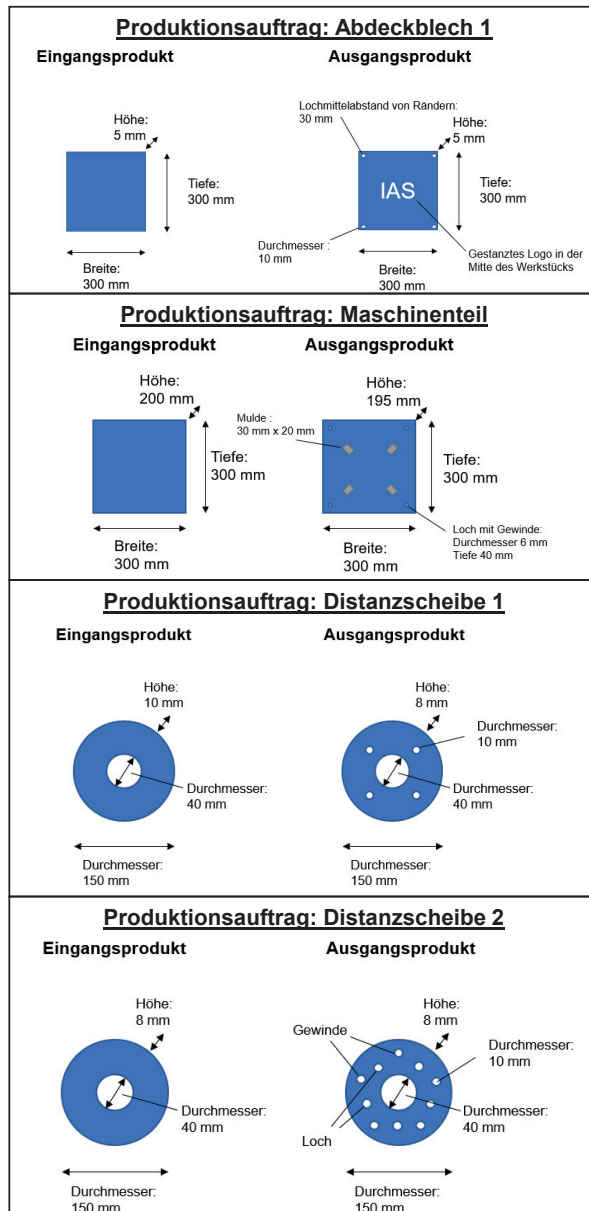


Abbildung 5.4: Übersicht von vier Produktionsaufträgen der Evaluierung

- 3) Ablauf des Rekonfigurationsmanagements mit Ergebnisausgabe: Das Rekonfigurationsmanagement läuft und findet eine geeignete Lösung; wenn eine Rekonfiguration erforderlich ist, wird das Ergebnis im XML-Format bereitgestellt (siehe Beispiel in Abbildung 5.5). Diese XML-Datei wird dabei wie in Kapitel 4.1 beschrieben vom Evaluation Agent ausgegeben und ermöglicht die Produktion mit der neu gefundenen Systemkonfiguration auf Basis des Produktionsauftrags. Sie enthält die CPPMs in ihrer jeweiligen Konfiguration und die Positionen der CPPMs im Layout. Im abgebildeten Beispiel befindet sich etwa ein Fräsmodule (*CPPMFrasen0*) in seiner Modulkonfiguration *FrasenA* an Position 3. Anschließend wird die Servicesequenz für die einzelnen Produktionsschritte angegeben, wobei neben den optimierten Produktionsparametern auch nicht variable Parameter aufgeführt werden, die durch den geforderten Produktionsauftrag vorgegeben sind. Im Beispiel wird der erste Produktionsschritt durch den Service *MetallLochBohrenD* durchgeführt und der optimierte Wert für den Produktionsparameter *Drehzahl* (*rotationalSpeed*) ist mit 1000 Umdrehungen pro Minute angegeben. Weiterhin werden als nicht variable Parameter eine *Bohrloch-Tiefe* (*depth*) von 40 mm und eine *Bohrloch-Anzahl* (*quantity*) von 4 vorgegeben. Danach ist der Gesamtaufwand, d.h. die Summe aus Rekonfigurations- und Produktionsaufwand für die Kriterien Zeit, Energie und Kosten zusammen mit der jeweils eingestellten Gewichtung eingetragen. Im Beispiel wurde für das Kriterium Zeit mit einer Gewichtung von 0,5 ein Aufwand von 358,83 Sekunden, für das Kriterium Energie mit einer Gewichtung von 0,25 ein Aufwand von 3,77 kWh und für das Kriterium Kosten mit einer Gewichtung von 0,25 ein Aufwand von 114,72 € ermittelt. Zuletzt werden zusätzlich der berechnete Nutzwert, das geforderte Auftragsvolumen und die Taktzeit, mit der die Eingangsprodukte in das Produktionssystem eintreten, in Millisekunden aufgeführt.

```

<Evaluation>
<CPMPPosition>
  <CPPM Name="CPPMFrasen0" Konfiguration="FrasenA">7</CPPM>
  <CPPM Name="CPPMBohren0" Konfiguration="BohrenC">15</CPPM>
  <CPPM Name="CPPMBohrenFrasen0" Konfiguration="BohrenFrasenB">19</CPPM>
</CPMPPosition>
<Produktionsschritte>
  <Produktionsschritt Service="MetallLochBohrenD" Nummer="1">
    <Parameter Name="rotationalSpeed">1000</Parameter>
    <Parameter Name="depth">40</Parameter>
    <Parameter Name="quantity">4</Parameter>
  </Produktionsschritt>
  <Produktionsschritt Service="MetallGewindeBohrenC" Nummer="2">
    <Parameter Name="rotationalSpeed">600</Parameter>
    <Parameter Name="depth">40</Parameter>
    <Parameter Name="quantity">4</Parameter>
  </Produktionsschritt>
  <Produktionsschritt Service="MetallFrasenA" Nummer="3">
    <Parameter Name="rotationalSpeed">1600</Parameter>
    <Parameter Name="length">300</Parameter>
    <Parameter Name="depth">5</Parameter>
  </Produktionsschritt>
  <Produktionsschritt Service="MetallMuldeFrasenC" Nummer="4">
    <Parameter Name="rotationalSpeed">2000</Parameter>
    <Parameter Name="length">50</Parameter>
    <Parameter Name="depth">5</Parameter>
  </Produktionsschritt>
</Produktionsschritte>
<Gesamtaufwand>
  <Aufwand Name="time" Gewichtung="0.5">358.83307</Aufwand>
  <Aufwand Name="energy" Gewichtung="0.25">3.772499</Aufwand>
  <Aufwand Name="cost" Gewichtung="0.25">114.71768</Aufwand>
</Gesamtaufwand>
<Nutzwert>0.9511084486323319</Nutzwert>
<Auftragsvolumen>2</Auftragsvolumen>
<Taktzeit>5000</Taktzeit>
</Evaluation>

```

Abbildung 5.5: Beispielhafte XML-Datei einer gefundenen alternativen Systemkonfiguration

- 4) Durchführung der Rekonfigurationsmaßnahmen: Die vorgeschlagenen Rekonfigurationsmaßnahmen werden durchgeführt, um die Transformation von der Startkonfiguration in die jeweilige neue Systemkonfiguration umzusetzen. Dieser Vorgang ist in Abbildung 5.6 visualisiert. Im Beispiel werden die rot umrandeten Produktionsmodule entfernt, das lila umrandete Produktionsmodul wird rekonfiguriert, das orange umrandete Produktionsmodul wird rekonfiguriert und seine Position wird geändert und das grün umrandete Produktionsmodul wird neu hinzugefügt. Das Tracking der Rekonfigurationsaufwände erfolgt dabei durch die in Kapitel 4.3 beschriebene Funktionalität des Monitorings der Unity-Simulation, die Übersicht der dazu genutzten Bedienelemente der Unity-GUI sind in Abbildung B.18 des Anhangs dargestellt.

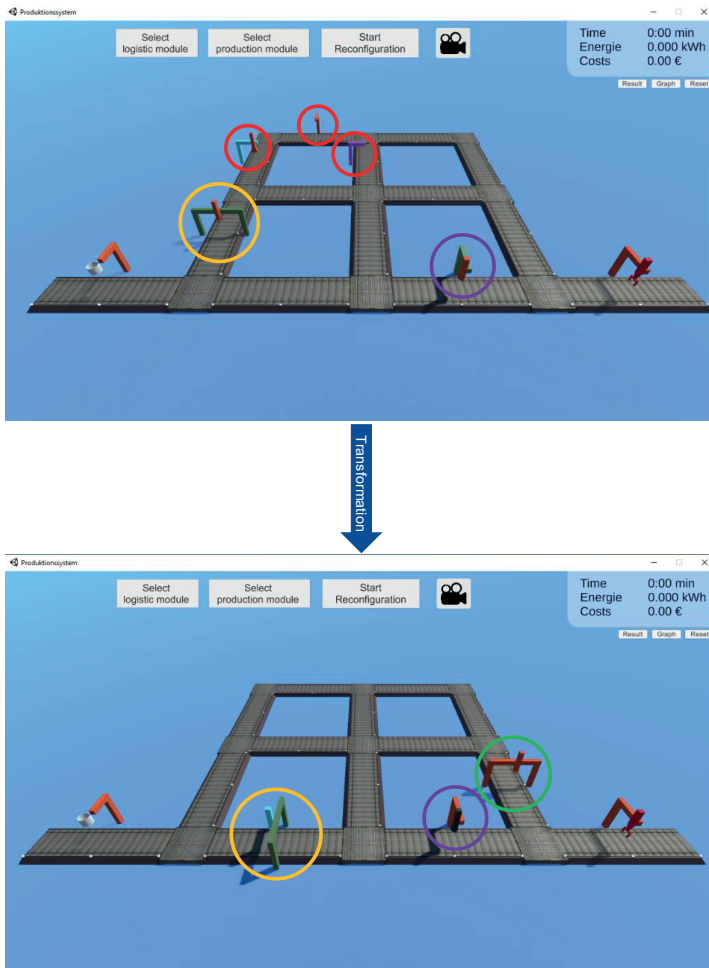


Abbildung 5.6: Beispielhafte Transformation einer Startkonfiguration in eine neue Systemkonfiguration

- 5) Durchführen der Produktion: Der Produktionsauftrag wird schließlich durch die Unity-Simulation und die SBC durchgeführt. Dazu wird wie in Kapitel 4.3 beschrieben der konkrete Produktionsauftrag im Sinne einer parametrisierten BOP, entsprechend dem Produktionsauftrag und auf Basis des vom Evaluation Agent gelieferten Ergebnisses im XML-Format, über das Webinterface der SBC (siehe Abbildung 4.20 in Kapitel 4.3) eingegeben.

- 6) Aufnehmen der Kennzahlen: Hierzu wird erneut die in Kapitel 4.3 beschriebene Monitoring-Funktionalität der Unity-Simulation genutzt. Dabei werden parallel zum Betrieb die Produktionsaufwände im Hinblick auf Zeit, Kosten und Energie berechnet und in der rechten oberen Ecke der Unity-GUI angezeigt. Nach Fertigstellung des Produktionsauftrags kann der dort befindliche Button „Result“ betätigt werden, um die in Schritt (4) ermittelten Rekonfigurationsaufwände mit den aufgenommenen Produktionsaufwänden zu den kriterienspezifischen Gesamtaufwänden aufzuaddieren. Daraufhin werden die Gesamtaufwände wie in Abbildung 5.7 dargestellt ausgegeben. Durch einen Vergleich mit den im Rahmen des Rekonfigurationsmanagements ermittelten Aufwänden kann die Vorhersagegenauigkeit der Aufwände durch das Rekonfigurationsmanagements untersucht werden.

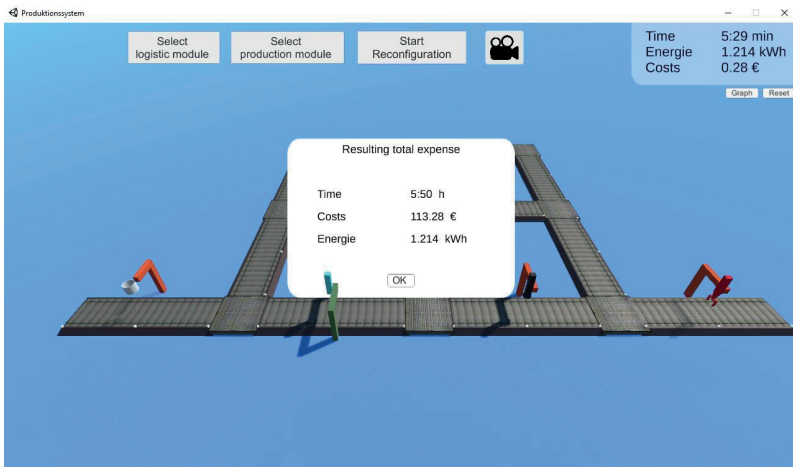


Abbildung 5.7: Beispiel für die resultierenden Gesamtaufwände nach Abschluss der Produktion in Unity

## 5.4 Auswertung der Ergebnisse der Evaluierungsszenarien

Die Ergebnisse der Auswertung der Evaluierungsszenarien sind in Tabelle 5.2 zusammengefasst. Es konnte festgestellt werden, dass die Durchführung aller zehn Evaluierungsszenarien zu korrekten Ergebnissen geführt hat. Im Durchschnitt ergab sich eine Genauigkeit von 92,3 % in Bezug auf die Vorhersage des gesamten Zeit-, Kosten- und Energieaufwands durch das Rekonfigurationsmanagement im Vergleich zu den durch die Monitoring-Funktionalität der Unity-Simulation ermittelten Werten. Es gibt also lediglich eine geringe Abweichung zwischen den prognostizierten Aufwänden und denen des Monitorings. Hierzu gilt für diese Forschungsarbeit, aber auch im Allgemeinen der Grundsatz, dass jegliche Methoden bzw. Algorithmen nur so gut sein können wie

das zugrundeliegende Wissen auf dem diese beruhen. Es besteht also eine große Abhängigkeit dazu, wie gut die Modellierung die Realität abbildet.

**Tabelle 5.2: Ergebnisse der Evaluierungsszenarien**

| #  | Produktionsauftrag | Rekonfigurationsbedarf<br>vorhanden? | Ergebnis<br>korrekt? | Genauigkeit |
|----|--------------------|--------------------------------------|----------------------|-------------|
| 1  | Abdeckblech 1      | Ja                                   | ✓                    | 85,90%      |
| 2  | Distanzscheibe 1   | Ja                                   | ✓                    | 97,60%      |
| 3  | Distanzscheibe 2   | Ja                                   | ✓                    | 98,50%      |
| 4  | Abdeckblech 2      | Ja                                   | ✓                    | 98,72%      |
| 5  | Abdeckblech 2      | Ja                                   | ✓                    | 96,56%      |
| 6  | Maschinenteil      | Ja                                   | ✓                    | 82,52%      |
| 7  | Maschinenteil      | Ja                                   | ✓                    | 79,03%      |
| 8  | Abdeckblech 1      | Ja                                   | ✓                    | 96,68%      |
| 9  | Distanzscheibe 2   | Ja                                   | ✓                    | 95,19%      |
| 10 | Abdeckblech 2      | Nein                                 | ✓                    | -           |

Zusammenfassend ergab die Evaluierung, dass das CPPS, da es mit einem selbstorganisierten Rekonfigurationsmanagement ausgestattet ist, in der Lage war, festzustellen, ob eine Rekonfiguration erforderlich war oder nicht. Wenn dies der Fall war, konnte es eine geeignete alternative Systemkonfiguration sowie eine entsprechende BOP bereitstellen, um die geforderte Produktion durchzuführen. Durch den realisierten Prototyp und die definierten Evaluierungsszenarien wurde dies in einem Evaluationsumfeld demonstriert, das insgesamt ein breites Spektrum abdeckt und sowohl eine hohe als auch variable Komplexität aufweist.

## 5.5 Abgleich mit den definierten Anforderungen an die Konzeption

Wie bereits in Kapitel 1 angekündigt wurde, erfolgt an dieser Stelle eine Beurteilung ob die formulierte Forschungsfrage hinreichend beantwortet wurde. Dies ist einerseits im DSR vorgesehen und dem Relevanz Zyklus zugeteilt (s. Kapitel 1.4), andererseits kann darüber der Wahrheitsgehalt der in Kapitel 1.3 aufgestellten These, dass CPPS vielversprechende Potenziale bieten um ein Rekonfigurationsmanagement durchführen zu können, untermauert werden.



Um die Erfüllung der definierten Anforderungen an die Konzeption zu gewährleisten wurde während der gesamten Konzeption und Realisierung zielgerichtet auf eben diese Erfüllung hingearbeitet. Die definierten Anforderungen haben demnach die Konzeption und Realisierung maßgeblich gestaltet. Die in diesem Kapitel vorgestellte Evaluierung zeigt durch die Funktionsfähigkeit des Prototypen gleichzeitig die Erfüllung der gestellten Anforderungen. Nachfolgend wird aufgezeigt durch welche Bestandteile des Konzepts (bzw. des entsprechenden Artefakts) die jeweiligen Anforderungen (A) erfüllt wurden.

Die Erfüllung von A1: *Geeignete Abbildung der Schritte des Rekonfigurationsmanagements im CPPS* wurde durch die Ausprägung der Methodik welche durch eben jene Schritte des Rekonfigurationsmanagements strukturiert ist bewerkstelligt. Durch diese Strukturierung wird der Detaillierung von Anforderung A1 in die Anforderungen *Ermittlung des Rekonfigurationsbedarfs* (A1.1), *Generierung alternativer Konfigurationen* (A1.2), *Bewertung der Konfigurationen* (A1.3) sowie *Auswahl einer Konfiguration* (A1.4) Rechnung getragen. Wie im Folgenden erläutert wird, sind diese Unter-Anforderungen erfüllt, wodurch auch A1 selbst erfüllt ist.

Die *Ermittlung des Rekonfigurationsbedarfs* (A1.1) wird im gleichnamigen Schritt der Methodik adressiert und durch den Abgleich der SOLL-Produktion mit der IST-Konfiguration umgesetzt. Die beiden auslösenden Trigger-Arten ‚Anforderungsänderung‘ und ‚Komponentenausfall‘ wurden im Zuge der präsentierten Evaluierung mehrfach erfolgreich evaluiert wodurch diese Anforderung als erfüllt bewertet wird.

Bezüglich der *Generierung alternativer Konfigurationen* (A1.2), sollte zunächst *auf der Maschinenebene die Hardware und Software umfasst* (A1.2.1) werden. Dies wurde innerhalb der *Generierung von Alternativen für Produktionssequenzen* sowie auch im zugehörigen Teil der Modellierung aufgegriffen. Durch die Evaluierung und die Betrachtung der vom Prototyp ausgegebenen Ergebnis-XML hat sich gezeigt, dass basierend auf den Modellen der CPPMs korrekte Produktionssequenzen gebildet werden. Dabei werden korrekte CPPM-Konfigurationsalternativen gewählt, welche über ihre Hardware- und Software-Komponenten definiert sind. Hierdurch wird A1.2.1 als erfüllt bewertet.

Außerdem soll *auf der Systemebene die Auswahl von Maschinen, deren Positionierung, sowie die Anpassung des Produktionsprozesses (also dessen Sequenz, Allokation und Parametrierung) umfasst werden* (A1.2.2). Dies wurde durch die entsprechende Ausprägung des Schrittes *Generierung alternativer Konfigurationen* und dessen Teilschritte sowie wiederum durch die zugehörige Modellierung realisiert. Die Evaluierung zeigt auf das die Maschinen über die Ausgabe von deren CPPM-Konfigurationsalternative ausgewählt, sowie deren Positionierung im Layout über Angabe der Position gegeben ist. Die *Anpassung des Produktionsprozesses* ist über die Ausgabe der Produktionsschritte und deren Reihenfolge (Sequenz) sowie deren Parametrierung, welche aus der

simulationsbasierten Optimierung resultiert, gegeben. Durch den Einsatz der SBC ist eine dynamische Allokation (und insbesondere auch Re-Allokation) der Services zu den Produktionsmodulen gegeben, weshalb diese zur Laufzeit und mit Hilfe des Discovery Servers geschieht. Durch die Abbildung der Steuerungslogik innerhalb des Simulationsmodells ist dieses Verhalten jedoch nachgebildet und somit implizit über die Beschreibung der Produktionsschritte in der Ergebnis-XML gegeben. Bei Bedarf könnte die konkrete Allokation der Services zu den jeweiligen CPPMs einfach mit ausgegeben werden. Die Anforderung A1.2.2 ist somit erfüllt.

Desweiteren sollte eine *Intelligente Exploration des Lösungsraums von Konfigurationsalternativen realisiert werden* (A1.2.3). Auch diese Anforderung wurde durch die beschriebene Ausprägung des Schrittes *Generierung alternativer Konfigurationen* und dessen Teilschritte realisiert. Durch den dezentralen Ansatz wird der Lösungsraum auf intelligente Art und Weise aufgeteilt und reduziert. Die jeweiligen Systemkonfigurations-Agenten bilden die Teil-Lösungsräume ab und prägen sich selbstorganisiert weiter aus, dabei wird auch eine gezielte Reduktion des Lösungsraums über die Einstellungen in der GUI ermöglicht. Die Evaluierung hat gezeigt das die einzelnen System Configuration Agenten jeweils korrekte alternative Systemkonfigurationen bilden welche (auch) über die Ergebnis-XML beschrieben werden. Außerdem kann der Lösungsraum über die GUI-Einstellungen gezielt reduziert werden. Die Anforderung A1.2.3 ist somit erfüllt.

Durch die Erfüllung der Anforderungen A1.2.1, A1.2.2 und A1.2.3 ist auch die Anforderung A1.2 erfüllt.

Die *Bewertung der Konfigurationen* (A1.3) sollte zunächst *den Rekonfigurationsaufwand berücksichtigen* (A1.3.1). Hierzu wird in der *Generierung von Alternativen für Produktionssequenzen* bereits der Rekonfigurationsaufwand auf der *Maschinenebene* ermittelt. Anschließend wird der Rekonfigurationsaufwand auf der *Systemebene* im Zuge der *Ermittlung von Layoutvarianten der Alternativen* berechnet. Der gesamte Rekonfigurationsaufwand wird dann bei der Bewertung und Auswahl einer Konfiguration wie in der Methodik (s. Kapitel 3.2) beschrieben berücksichtigt. Die benötigten Informationen werden über die aus der Modellierung (s. Kapitel 3.3) resultierenden Modelle bereitgestellt.

Um auch *den Produktionsaufwand berücksichtigen* (A1.3.2) zu können wurden die diesbezüglichen Zusammenhänge entsprechend modelliert (s. Kapitel 3.3). Die Produktionsaufwände können somit im Zuge der *Optimierung der Produktionsparameter der Produktionsschritte* optimiert werden. Anschließend werden auch diese optimierten Produktionsaufwände bei der Bewertung und Auswahl einer Konfiguration berücksichtigt.

Zudem soll die Bewertung der alternativen Konfigurationen *multikriteriell und anhand objektiver Kriterien erfolgen* (A1.3.3). Hierzu wurden sowohl die Rekonfigurations- als auch die Produktionsaufwände zunächst multikriteriell modelliert (s. Kapitel 3.3) und bereits im Schritt der *Generierung alternativer Konfigurationen* multikriteriell berechnet, bzw. optimiert (s. Kapitel 3.2).

Anschließend erfolgen auch die Bewertung und Auswahl einer Konfiguration multikriteriell anhand der objektiven Kriterien (Zeit, Kosten und Energie).

Somit sind die Anforderungen A1.3.1, A1.3.2 und A1.3.3 erfüllt, wodurch auch A1.3 erfüllt ist. Die resultierenden Gesamtaufwände für die Produktion des jeweiligen Produktionsauftrags ist in der Ergebnis-XML für die jeweilige alternative Systemkonfiguration angegeben.

Die geforderte *Auswahl einer Konfiguration* (A1.4) ist im Zuge der Beurteilung von A1.3 bereits mehrfach angesprochen worden. Dadurch ist deren Erfüllung bereits ersichtlich. Zudem rundet die jeweilige Ausgabe der Ergebnis-XML (zweckmäßige Beschreibung der Systemkonfiguration) für die am besten geeignete Systemkonfiguration, also diejenige mit dem höchsten Nutzwert, die Erfüllung dieser Anforderung ab. Die Evaluierung anhand der zehn Evaluierungsszenarien zeigte sowohl die korrekte Berechnung der Nutzwerte, als auch die korrekte Auswahl basierend auf diesem Nutzwert auf. Die Anforderung A1.4 ist somit erfüllt.

Die *Automatisierte Durchführung des Rekonfigurationsmanagements* (A2) ist dadurch umgesetzt, dass die beschriebene CPPS-Architektur die entwickelte Methodik voll automatisiert bzw. autonom durchführt. Um diese Anforderung bewerten zu können wurde wie auch bereits in Kapitel 2.1 eine Einstufung des Level of Automation (LoA) nach [92] vorgenommen. Durch die autonome Durchführung des Rekonfigurationsmanagements durch das CPPS selbst (ohne menschliche Partizipation), welche über die Evaluierung aufgezeigt wurde, wird ein LoA von 10 (Maximum) erreicht. Die Anforderung A2 ist somit erfüllt.

Die *Nutzung der Potenziale von cyber-physischen Produktionssystemen* (A3) ist über die Nutzung der Modelle, welche auf der beschriebenen Modellierung (s. Kapitel 3.3) beruhen, sowie über die Intelligenz der Softwareagenten, welche die Methodik umsetzen, und deren dezentrale Vernetzung innerhalb der CPPS-Architektur, gegeben. Die Evaluierung hat aufgezeigt, dass der realisierte Prototyp, welcher die zuvor genannten Aspekte umsetzt, für ein breites Spektrum an Evaluierungsszenarien funktionsfähig ist. Die Anforderung A3 ist somit erfüllt.

Zusammengefasst sind somit alle definierten Anforderungen an die Konzeption erfüllt worden. Demnach wurde die formulierte Forschungsfrage:

*Wie können cyber-physische Produktionssysteme um die Fähigkeit eines selbstorganisierten Rekonfigurationsmanagements bereichert werden?*

durch die vorliegende Forschungsarbeit hinreichend beantwortet. Darüber hinaus konnte der Wahrheitsgehalt der in Kapitel 1.3 aufgestellten These, dass CPPS vielversprechende Potenziale bieten um ein Rekonfigurationsmanagement durchführen zu können, untermauert werden.

## 6 Schlussbetrachtung

Dieses Kapitel schließt die Forschungsarbeit mit einer Zusammenfassung der Ergebnisse und einem Ausblick auf mögliche weitere Forschungsaktivitäten.

### 6.1 Zusammenfassung der Ergebnisse

Kürzere Produktlebenszyklen und die zunehmende Individualisierung der Produktion führen zu einem erhöhten Rekonfigurationsbedarf in der Domäne der industriellen Automatisierung, die in Zukunft von cyber-physischen Produktionssystemen dominiert werden wird. Obwohl die Notwendigkeit der Rekonfiguration von Produktionssystemen unumstritten ist, wie diese bisher nur selten realisiert. Geschuldet ist dies der Tatsache, dass diese gegenwertigen Defizite wie Fehleranfälligkeit aufgrund von fehlender systematischer und methodischer Unterstützung sowie einen hohen zeitlichen Aufwand aufgrund manueller Planung und Durchführung aufweist. Zudem besteht eine hohe Abhängigkeit von menschlichem Wissen und Erfahrungswerten, was zu einer nicht objektiven und suboptimalen Rekonfiguration führt. Diese Defizite wurden daher im Rahmen dieser Forschungsarbeit mit Hilfe eines Rekonfigurationsmanagements adressiert. Zudem wurde die These formuliert, dass Cyber-physische Produktionssysteme vielversprechende Potenziale bieten, um ein solches Rekonfigurationsmanagement durchführen zu können. Insbesondere da sie über Modelle verfügen, Intelligenz bieten können und vernetzt sind.

Die Analyse des Stands der Wissenschaft und Technik hat gezeigt, dass sich zahlreiche Forschungsaktivitäten mit Teilaspekten des betrachteten Forschungsgegenstandes befassen, was das die Relevanz der Thematik unterstreicht. Jedoch wurde über eine Untersuchung des Erfüllungsgrades der abgeleiteten Anforderungen an die Konzeption aufgezeigt, dass die formulierte Forschungsfrage: *Wie können cyber-physische Produktionssysteme um die Fähigkeit eines selbstorganisierten Rekonfigurationsmanagements bereichert werden?* nicht hinreichend beantwortet wird.

In dieser Dissertation wurde daher ein Konzept entwickelt mit dem cyber-physische Produktionssysteme um die Fähigkeit zu einem selbstorganisierten Rekonfigurationsmanagement bereichert werden können.

Das Konzept besteht aus der Entwicklung einer Methodik, einer Modellierung des für die Methodik erforderlichen Wissens und einer CPPS-Architektur zur Integration von Methodik und Modellierung.

Die **Methodik** setzt dabei die **vier** identifizierten **Schritte des Rekonfigurationsmanagements** um, beginnend mit der *Ermittlung des Rekonfigurationsbedarfs* unter Verwendung eines **Fähig-**

**keitsmodells** des cyber-physischen Produktionssystems, das auf der **Formalisierten Prozessbeschreibung** basiert, um den funktionalen Abgleich mit einem Produktionsauftrag zu erlauben. Zeigt dieser auf, dass eine Rekonfiguration benötigt wird, erfolgen die nachgestellten Schritte der *Generierung alternativer Konfigurationen*, der *Bewertung der Konfigurationen* und der *Auswahl einer Konfiguration*. **Dabei wird der Lösungsraum an alternativen Systemkonfigurationen systematisch und vollautomatisiert in einem intelligenten, dezentralen Verfahren exploriert, welches geeignete Systemkonfigurationen ermittelt und anschließend simulationsbasiert, multikriteriell (Zeit, Kosten und Energie) optimiert.** Danach erfolgt eine Bewertung der alternativen Systemkonfigurationen mittels einer **Nutzwertanalyse** basierend auf den **Rekonfigurations- und Produktionsaufwänden** und die geeignetste alternative Konfiguration wird ausgewählt und kann anschließend auf das Produktionssystem übertragen werden.

Die **Modellierung**, die insbesondere die Produktionsaufträge und die cyber-physischen Module umfasst, beruht auf der Verwendung von UML/XML. Um einen funktionalen Abgleich zu ermöglichen, basiert die Modellierung auf dem **PPR-Konzept** und der **Formalisierten Prozessbeschreibung**. Darüber hinaus werden die unterschiedlichen Modulkonfigurationen auch hinsichtlich ihrer **Rekonfigurations- und Produktionsaufwände** beschrieben und insbesondere werden **Zustände modelliert**, um die simulationsbasierte Optimierung der Produktionsparameter zu ermöglichen.

Die **CPPS-Architektur** integriert sowohl die Wissensmodellierung und das Wissensmanagement, als auch die Methodik und gliedert sich in vier Schichten: **Management-Layer, Proxy-Layer, Control-Layer** und **Asset-Layer**. Wesentliche Merkmale der Architektur sind, dass für die Steuerung des Asset-Layers im Control-Layer eine **service-orientierte Architektur** zum Einsatz kommt, während im Management-Layer die konzipierte Methodik durch den Einsatz eines **Multiagentensystems** umgesetzt wird.

Die **prototypische Implementierung** des Konzeptes wurde anhand der vier CPPS-Architektur-Layer beschrieben und zeigt insbesondere, wie die in **Java** mit dem Java Agent Development Framework entwickelten **Softwareagenten** des Management-Layers die konzipierte Methodik umsetzen und dabei die Modelle des Proxy-Layers nutzen. Insbesondere sind die Verfahrensvarianten zur Ermittlung von Layoutvarianten basierend auf einem **genetischen Algorithmus** sowie die simulationsbasierte, multikriterielle Optimierung der Systemkonfigurationen beschrieben. Für letztere erfolgt die **dynamische Erstellung des zugehörigen Simulationsmodells in MATLAB Simulink** zur Laufzeit mit Hilfe von **parallel ausgeführten Zustandsautomaten** im Rahmen einer **diskreten ereignisorientierten Simulation**. Die anschließende simulationsbasierte, multikriterielle Optimierung der Produktionsparameter wurde anhand des realisierten **multikriteriellen Simulated Annealing Algorithmus**, als eines von mehreren wählbaren Optimierungsverfahren, erläutert. Die Realisierung des Control- und des Asset-Layers erfolgte durch ein in **Unity**

**simuliertes modulares Produktionssystem mit Matrix-Layout**, das durch ein **OPC UA Steuerungsnetzwerk** service-orientiert gesteuert wird und im Zuge der Evaluierung eingesetzt wurde.

Die durchgeführte empirische Evaluierung zeigt basierend auf der beschriebenen Realisierung die Erfüllung der Anforderungen an die Konzeption auf. Insbesondere konnten dadurch sowohl die formulierte Forschungsfrage als hinreichend beantwortet, als auch die eingangs aufgestellte These als untermauert bewertet werden. Dazu wurde der systematische Ablauf der Evaluierung vorgestellt, nach dessen Muster zehn Evaluierungsszenarien definiert, durchgeführt und ausgewertet wurden. Dabei wurde ein breites Spektrum an Evaluierungsszenarien untersucht, welche die Funktionalität für verschiedene Trigger-Arten (neue Produktionsaufträge sowie Komponentenausfälle), variierende Komplexität, unterschiedliche Start-Konfigurationen sowie diverse Produktionsaufträge aufzeigen konnten. Mit Hilfe der realisierten Monitoring-Fähigkeit der Unity-Simulation konnte zudem eine durchschnittliche Vorhersage-Genauigkeit von 92,3 % für den gesamten Zeit-, Kosten- und Energieaufwand durch das Rekonfigurationsmanagement aufgezeigt werden. Somit wurde dargestellt, dass das cyber-physische Produktionssystem nun in der Lage ist Rekonfigurationsbedarf zur Betriebszeit zu erkennen und selbstorganisiert eine geeignete Konfigurationsoption für das individuell vorliegende Szenario zu finden.

Als eingesetzte Forschungsmethodik wurde das Design Science Research gewählt. So wurden gemäß dem Relevanz Zyklus die Problemstellung identifiziert, die Forschungsfrage formuliert und die Anforderungen an das Konzept als wesentliches, zu entwerfendes Artefakt abgeleitet, die auch die Grundlage der Evaluierung bilden. Der iterative Prozess des Entwurfs im Sinne des Design Zyklus wurde hingegen eingesetzt, um sowohl das wesentliche Artefakt durch die Ausprägung der drei Teilartefakte des Konzepts, also Methodik, Modellierung und CPPS-Architektur, als auch die Realisierung des Konzepts im Sinne eines zweiten Artefakts systematisch zu erarbeiten. Der resultierende Prototyp bietet daher ein hohes Maß an Flexibilität und kann anwendungsspezifisch konfiguriert werden, insbesondere können verschiedene Optimierungsverfahren wie zum Beispiel verschiedene Varianten des Simulated Annealing oder des Genetischen Algorithmus gewählt werden. Im Rahmen des Rigor Zyklus wurden sowohl die bestehende Forschungslücke als auch Ansätze für den späteren Entwurf der definierten (Teil-)Artefakte basierend auf der existierenden Wissensbasis im Sinne des Stands der Wissenschaft und Technik identifiziert. Um die existierende Wissensbasis auch in verallgemeinerter Form bereichern zu können, wurde für die Beschreibung der intelligenten Exploration von Lösungsräumen, die bei der *Generierung alternativer Konfigurationen* zum Einsatz kommt, der abstrakte Begriff der *Filter* eingeführt. So dient die beschriebene rekonfigurationsmanagementspezifische Ausprägung zugleich als beispielhafte Umsetzung einer universellen Methodik zur intelligenten Exploration von Lösungsräumen, die zudem im Rahmen dieser Forschungsarbeit in [154] veröffentlicht wurde.

## 6.2 Ausblick

Im Zuge dieser Forschungsarbeit wurden außerdem die folgenden Anknüpfungspunkte als vielversprechend im Sinne einer Erweiterung oder Verbesserung der vorgestellten Arbeit identifiziert:

**Transfer in die industrielle Anwendung:** Im Rahmen dieser Forschungsarbeit wurde aufgrund der Verfügbarkeit und des Aufwandes sowohl ein simuliertes Produktionssystem verwendet, als auch eine Modellierung die insbesondere aus der Betrachtung der Literatur resultiert. Dabei wurde zwar auf eine ausreichende Realitätsnähe und Komplexität geachtet, um eine aussagekräftige Evaluierung durchführen zu können, dennoch verspricht der Transfer in die industrielle Anwendung weiteres Verbesserungspotenzial. Darüber hinaus wird erwartet, dass die hohe Anpassbarkeit und Flexibilität des modularen Konzepts sowie der agentenbasierten Implementierung ein einfaches Tailoring auf die spezifischen Gegebenheiten der industriellen Anwendung ermöglicht. So kann beispielsweise eine aufwandsarme Anpassung hinsichtlich spezifisch relevanter Zielkriterien erfolgen, wenn die bestehende Betrachtung von Zeit, Kosten und Energie nicht favorisiert wird. Zudem wurde für die funktionale Modellierung die interface-orientierte Formalisierte Prozessbeschreibung gewählt, um ein gewünschtes Produkt zu beschreiben, was zu einem erhöhten Freiheitsgrad durch die Generierung von alternativen Produktionssequenzen führt. Während dieser Ansatz im Rahmen der vorliegenden Forschungsarbeit favorisiert wurde, so ist es in der industriellen Anwendung durchaus denkbar, dass eine vorgegebene BOP zugrunde gelegt wird. Auch hier erlaubt der modulare Charakter des Konzepts eine aufwandsarme Anpassung, die an dieser Stelle zudem zu einer Komplexitätsreduktion führt.

**Einsatz einer vollwertigen Ontologie:** Da die Anforderungen, die das selbstorganisierte Rekonfigurationsmanagement an die Wissensrepräsentation stellt, wie beschrieben durch den einen UML/XML-basierten Ansatz erfüllt werden, wurde dieser Ansatz anstelle der aufwändigeren Verwendung einer vollwertigen Ontologie gewählt. Nichtsdestotrotz könnte die Anwendung einer vollwertigen Ontologie in mancher Hinsicht einen zusätzlichen Nutzen bringen, der über die Anforderungen des gegebenen Anwendungsfalls hinausgeht. Dies wird durch die höhere semantische Ausdruckskraft, die Möglichkeit, die Modellkomplexität zu erhöhen, die Inferenz und Aspekte der konstanten Evolution erreicht, die zu zusätzlichen Mehrwerten führen, wie etwa:

- Das Erreichen einer semantischen Interoperabilität, die aufgrund einer höheren semantischen Ausdruckskraft ein noch höheres Maß an semantischer Heterogenität bewältigen kann. Dies kann durch ein sogenanntes Ontologie-Matching, genauer gesagt durch Ontologie-Merging oder Ontologie-Reconciliation, erreicht werden, was zu einem geeigneten Alignment führt.
- Der Wechsel von einer Closed-World- hin zu einer Open-World-Assumption, ermöglicht durch die Option einer dynamischen Evolution des Informationsmodells während der Laufzeit. Zum Beispiel, um Kontextbewusstsein zu erreichen, d.h. um mit dynamischen

Änderungen während der Laufzeit umgehen zu können, die sogar die Semantik der Ontologie beeinflussen könnten. Auf systeminterner Ebene könnten fehlende wechselseitige Abhängigkeiten zwischen Systemkomponenten hinzugefügt oder bestehende geändert werden. Auf der externen Ebene kann die Kopplung oder Entkopplung zwischen kooperierenden Systemen einbezogen werden. Auf diese Weise kann die Entscheidungsfindung kontextbewusst durchgeführt werden.

- Die Erstellung einer geeigneten Wissensrepräsentation für mehr als nur eine Anwendung, aufgrund ihrer höheren semantischen Ausdruckskraft und der Abdeckung spezifischer Domänen, was für viele industriell genutzte CPPS notwendig sein wird.
- Die Inferenz kann zur Gewinnung neuer Erkenntnisse oder zur Verbesserung der Modellqualität (im Sinne der semantischen Aussagekraft) genutzt werden.

**Integration von Industriestandards:** Ein weiterer Aspekt ist die Integration von Industriestandards wie ISA-88 (zur Beschreibung von Zustandsautomaten). So kann Interoperabilität gewährleistet werden und potenziell eine höhere Akzeptanz im industriellen Umfeld erreicht werden. Dieser Aspekt kann auch in Verbindung mit der oben genannten Verwendung einer vollwertigen Ontologie adressiert werden. Hierzu kann beispielsweise die in [111] vorgeschlagene Methodik zur Entwicklung von Ontologien angewendet werden, die auf dem Einsatz von Ontology Design Pattern (ODPs) zur Integration von Industriestandards basiert. Es wird davon ausgegangen, dass der in dieser Forschungsarbeit vorgeschlagene Modellierungsansatz leicht anpassbar ist, da seine Entwicklung bereits durch mehrere Standards inspiriert wurde. Dabei können bestehende ODPs so weit wie möglich genutzt und gegebenenfalls erweitert werden. Insbesondere im Hinblick auf die in dieser Forschungsarbeit verwendeten nicht-funktionalen Aspekte wird erwartet, dass ein neues oder zumindest angepasstes ODP benötigt wird. Da mit dem beschriebenen UML-Klassendiagramm bereits eine Lightweight Ontology verfügbar ist, ist ein erster Schritt in diese Richtung bereits getan.

Andererseits sind die bereits erwähnte Verbreitung einheitlicher OPC UA Companion Specifications sowie standardisierter VWS Teilmodelle vielversprechende Entwicklungen. Wobei die Anforderungen des selbstorganisierten Rekonfigurationsmanagements nach bestem Wissen noch nicht vollständig bedient werden. Generell kann das Konzept der VWSs als alternative Möglichkeit genutzt werden, um eine standardisierte Wissensrepräsentation zu erreichen. Darüber hinaus können VWSs sowohl mit einem UML/XML-basierten als auch mit einem Ontologie-basierten Ansatz kombiniert werden, wodurch sich mehrere Richtungen für mögliche zukünftige Arbeiten ergeben.

**Generierung neuer Fertigungsservices zur Betriebszeit:** Es besteht die Möglichkeit, eine Generierung neuer Fertigungsservices durch die Intelligenz der einzelnen cyber-physischen Produk-



tionsmodule zu realisieren. Diese könnten über die bestehenden Services der vorhandenen alternativen Modulkonfigurationen hinaus zusätzlich bedarfsorientiert Funktionalität zur Betriebszeit erschließen. Denkbar sind hier beispielsweise auf Reinforcement Learning basierende Ansätze zur zielgerichteten Erzeugung von Steuerungsabläufen eines Produktionsmoduls. Da eine solche zusätzliche Intelligenz der cyber-physischen Produktionsmodule bereits bei der Konzeptentwicklung berücksichtigt wurde, können die so neu erschlossenen Fertigungsservices nahtlos in die *Generierung alternativer Konfigurationen* des Rekonfigurationsmanagements integriert werden.

## 7 Literaturverzeichnis

- [1] Verein Deutscher Ingenieure, “VDI/VDE 2653-1, Agentensysteme in der Automatisierungstechnik - Grundlagen,” 2010.
- [2] DIN SPEC, “91345: 2016-04 Reference Architecture Model Industrie 4.0 (RAMI4. 0),” *Berlin, Germany*, Bd. 10, Nr. 2436156, S. 10–31030, 2016.
- [3] G. Lanza, P. Nyhuis, J. Fisel, A. Jacob, L. Nielsen, und M. Schmidt *et al.*, “Wandlungsfähige, menschenzentrierte Strukturen in Fabriken und Netzwerken der Industrie 4.0,” *acatech Studie München*,” *Herbert Utz Verlage*, 2018.
- [4] E. Geisberger und M. Broy, *agendaCPS: Integrierte Forschungsagenda Cyber-Physical Systems*: Springer-Verlag, 2012.
- [5] A. C.L. Fipa, “Fipa acl message structure specification,” *Foundation for Intelligent Physical Agents*, <http://www.fipa.org/specs/fipa00061/SC00061G.html> (30.6. 2004), 2002.
- [6] L. Scrucca, “GA: A package for genetic algorithms in R,” *Journal of Statistical Software*, Bd. 53, S. 1–37, 2013.
- [7] D. Bakken, “Middleware,” *Encyclopedia of Distributed Computing*, Bd. 11, 2001.
- [8] M. Gharbi, A. Koschel, A. Rausch, und G. Starke, *Basiswissen für Softwarearchitekten: Aus-und Weiterbildung nach iSAQB-Standard zum Certified Professional for Software Architecture-Foundation Level*: dpunkt. verlag, 2020.
- [9] Verein Deutscher Ingenieure, “VDI/VDE 3633-1, Simulation von Logistik-, Materialfluss- und Produktionssystemen - Grundlagen,” 2014.
- [10] J. Matevska, “Rekonfiguration komponentenbasierter Softwaresysteme,” in *Rekonfiguration komponentenbasierter Softwaresysteme zur Laufzeit*: Springer, 2010, S. 73–81.
- [11] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, und G. Ulsoy *et al.*, “Reconfigurable manufacturing systems,” *CIRP annals*, Bd. 48, Nr. 2, S. 527–540, 1999.
- [12] A. F. Hees, *System zur Produktionsplanung für rekonfigurierbare Produktionssysteme*: Herbert Utz Verlag, 2017.
- [13] Y. Koren und S. Kota, “Reconfigurable machine tool,” 1999.
- [14] C. M. MacKenzie, K. Laskey, F. McCabe, P. F. Brown, R. Metz, und B. A. Hamilton, “Reference model for service oriented architecture 1.0,” *OASIS standard*, Bd. 12, S18, S. 1–31, 2006.
- [15] K. Gutenschwager, M. Rabe, S. Spieckermann, und S. Wenzel, *Simulation in Produktion und Logistik*: Springer, 2017.
- [16] D. Bertsimas und J. Tsitsiklis, “Simulated annealing,” *Statistical science*, Bd. 8, Nr. 1, S. 10–15, 1993.
- [17] M. Bortolini, E. Ferrari, F. G. Galizia, und C. Mora, “Implementation of reconfigurable manufacturing in the Italian context: State-of-the-art and trends,” *Procedia Manufacturing*, Bd. 39, S. 591–598, 2019.
- [18] E. Järvenpää, N. Siltala, und M. Lanz, “Formal resource and capability descriptions supporting rapid reconfiguration of assembly systems,” in *2016 IEEE International symposium on assembly and manufacturing (ISAM)*, 2016, S. 120–125.
- [19] I. Barros Garcia, J. Daaboul, A. Jouglet, und J. Le Duigou, “An approach to jointly optimize the process plan, scheduling, and layout design in reconfigurable manufacturing systems,” in *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, 2021, S. 403–415.

- [20] M. Paul, A. Cerqueus, D. Schneider, H. H. Benderbal, X. Boucher, und D. Lamy *et al.*, “Reconfigurable digitalized and servitized production systems: requirements and challenges,” in *Advances in Production Management Systems. Towards Smart and Digital Manufacturing: IFIP WG 5.7 International Conference, APMS 2020, Novi Sad, Serbia, August 30-September 3, 2020, Proceedings, Part II*, 2020, S. 501–508.
- [21] F. Boschi, G. Tavola, und M. Taisch, “A description and analysis method for reconfigurable production systems based on finite state automaton,” in *Service Orientation in Holonic and Multi-Agent Manufacturing: Proceedings of SOHOMA 2016*, 2017, S. 349–358.
- [22] F. Boschi, C. Zanetti, G. Tavola, M. Taisch, P. Leitão, und J. Barbosa *et al.*, “From key business factors to KPIs within a reconfigurable and flexible cyber-physical system,” in *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, 2017, S. 732–740.
- [23] J. Hermann, P. Rübel, M. Birtel, F. Mohr, A. Wagner, und M. Ruskowski, “Self-description of Cyber-Physical Production Modules for a product-driven manufacturing system,” *Procedia Manufacturing*, Bd. 38, S. 291–298, 2019.
- [24] S. Komesker, W. Motsch, J. Popper, A. Sidorenko, A. Wagner, und M. Ruskowski, “Enabling a multi-agent system for resilient production flow in modular production systems,” *Procedia CIRP*, Bd. 107, S. 991–998, 2022.
- [25] B. Vogel-Heuser, C. Diedrich, A. Fay, S. Jeschke, S. Kowalewski, und M. Wollschlaeger *et al.*, “Challenges for software engineering in automation,” *Journal of Software Engineering and Applications*, Bd. 2014, 2014.
- [26] A. Bildstein, J. Feng, und T. Bauernhansl, “Combining Channel Theory and Semantic Web Technology to build up a Production Capability Matching Framework,” *Procedia CIRP*, Bd. 81, S. 139–144, 2019.
- [27] J. Siegert, T. Schlegel, und T. Bauernhansl, “Matrix fusion factory,” *Procedia Manufacturing*, Bd. 23, S. 177–182, 2018.
- [28] C. Müller-Schloer und S. Tomforde, *Organic Computing-Technical Systems for Survival in the Real World*: Springer, 2017.
- [29] F. Karl, *Bedarfsermittlung und Planung von Rekonfigurationen an Betriebsmitteln*: Herbert Utz Verlag, 2015.
- [30] T. Müller, N. Jazdi, J.-P. Schmidt, und M. Weyrich, “Cyber-Physical Production Systems: enhancement with a self-organized reconfiguration management,” *Procedia CIRP*, Bd. 99, S. 549–554, 2021.
- [31] T. Müller, S. Walth, N. Jazdi, und M. Weyrich, “Identification of Reconfiguration Demand and Generation of Alternative Configurations for Cyber-Physical Production Systems,” in *Advances in Automotive Production Technology-Theory and Application*: Springer, 2021, S. 63–70.
- [32] T. Müller, B. Lindemann, T. Jung, N. Jazdi, und M. Weyrich, “Enhancing an Intelligent Digital Twin with a Self-organized Reconfiguration Management based on Adaptive Process Models,” *Procedia CIRP*, Bd. 104, S. 786–791, 2021.
- [33] N. Jazdi, “Cyber physical systems in the context of Industry 4.0,” in *2014 IEEE international conference on automation, quality and testing, robotics*, 2014, S. 1–4.
- [34] B. Bagheri, S. Yang, H.-A. Kao, und J. Lee, “Cyber-physical systems architecture for self-aware machines in industry 4.0 environment,” *IFAC-PapersOnLine*, Bd. 48, Nr. 3, S. 1622–1627, 2015.
- [35] B. Vogel-Heuser, M. Böhm, F. Brodeck, K. Kugler, S. Maasen, und D. Pantförder *et al.*, “Interdisciplinary engineering of cyber-physical production systems: highlighting the benefits of a combined interdisciplinary modelling approach on the basis of an industrial case,” *Design Science*, Bd. 6, 2020.

- [36] J. Wan, B. Yin, Di Li, A. Celesti, F. Tao, und Q. Hua, "An ontology-based resource reconfiguration method for manufacturing cyber-physical systems," *IEEE/ASME Transactions on Mechatronics*, Bd. 23, Nr. 6, S. 2537–2546, 2018.
- [37] M. Grochowski, H. Simon, D. Bohlender, S. Kowalewski, A. Löcklin, und T. Müller *et al.*, "Formale Methoden für rekonfigurierbare cyber-physische Systeme in der Produktion," *at-Automatisierungstechnik*, Bd. 68, Nr. 1, S. 3–14, 2020.
- [38] J. G. Represa und J. Delsing, "Autonomous production workstation operation, reconfiguration and synchronization," *Procedia Manufacturing*, Bd. 39, S. 226–234, 2019.
- [39] M. W. Waibel, L. P. Steenkamp, N. Moloko, und G. A. Oosthuizen, "Investigating the effects of smart production systems on sustainability elements," *Procedia Manufacturing*, Bd. 8, S. 731–737, 2017.
- [40] D. Stock, T. Bauernhansl, M. Weyrich, M. Feurer, und R. Wutzke, "System architectures for cyber-physical production systems enabling self-x and autonomy," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020, Bd. 1, S. 148–155.
- [41] C. E. Pereira, C. Diedrich, und P. Neumann, "Communication Protocols for Automation," in *Springer Handbook of Automation*: Springer, 2023, S. 535–560.
- [42] L. Monostori, B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, und G. Reinhart *et al.*, "Cyber-physical systems in manufacturing," *CIRP annals*, Bd. 65, Nr. 2, S. 621–641, 2016.
- [43] D. Pantförder, F. Mayer, C. Diedrich, P. Göhner, M. Weyrich, und B. Vogel-Heuser, "Agentenbasierte dynamische Rekonfiguration von vernetzten intelligenten Produktionsanlagen-Evolution statt Revolution," *Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung/periodcentered Migration*, S. 145–158, 2014.
- [44] K. D. Bettenhausen und S. Kowalewski, "Cyber-physical systems: Chancen und Nutzen aus Sicht der Automation," *VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik*, S. 9–10, 2013.
- [45] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*, 2008, S. 363–369.
- [46] L. Monostori, "Cyber-physical production systems: Roots, expectations and R&D challenges," *Procedia CIRP*, Bd. 17, S. 9–13, 2014.
- [47] T. Bauernhansl, M. Weyrich, L. Zarco, T. Müller, P. Marks, und T. Schlegel *et al.*, "Semantic structuring of elements and capabilities in ultra-flexible factories," *Procedia CIRP*, Bd. 93, S. 335–340, 2020.
- [48] M. G. Mehrabi, A. G. Ulsoy, und Y. Koren, "Reconfigurable manufacturing systems and their enabling technologies," *International Journal of Manufacturing Technology and Management*, Bd. 1, Nr. 1, S. 114–131, 2000.
- [49] C. Bettaieb, A. J. Telmoudi, A. Sava, und L. Nabli, "Reconfigurable manufacturing system: Overview and proposition of new approach," in *2017 International Conference on Control, Automation and Diagnosis (ICCAD)*, 2017, S. 534–539.
- [50] M. G. Mehrabi, A. G. Ulsoy, und Y. Koren, "Reconfigurable manufacturing systems: Key to future manufacturing," *Journal of Intelligent manufacturing*, Bd. 11, Nr. 4, S. 403–419, 2000.
- [51] A. Youssef und H. A. ElMaraghy, "Optimal configuration selection for reconfigurable manufacturing systems," *International Journal of Flexible Manufacturing Systems*, Bd. 19, Nr. 2, S. 67–106, 2007.
- [52] E. Järvenpää, N. Siltala, O. Hylli, und M. Lanz, "Capability matchmaking procedure to support rapid configuration and re-configuration of production systems," *Procedia Manufacturing*, Bd. 11, S. 1053–1060, 2017.
- [53] Y. Koren, X. Gu, und W. Guo, "Reconfigurable manufacturing systems: Principles, design, and future trends," *Frontiers of Mechanical Engineering*, Bd. 13, Nr. 2, S. 121–136, 2018.

- [54] L. Zarco, J. Siegert, und T. Bauernhansl, "Software Model Requirements Applied to a Cyber-Physical Modular Robot in a Production Environment," *Procedia CIRP*, Bd. 81, S. 352–357, 2019.
- [55] M. Weyrich, "Was ist Automatisierungstechnik? Grundbegriffe und Konzepte der Automatisierungstechnik," in *Industrielle Automatisierungs- und Informationstechnik: IT-Architekturen, Kommunikation und Software zur Systemgestaltung*: Springer, 2023, S. 11–41.
- [56] T. Stehle und U. Heisel, "Konfiguration und Rekonfiguration von Produktionssystemen," in *Neue Entwicklungen in der Unternehmensorganisation*: Springer, 2017, S. 333–367.
- [57] F. Karl, *Bedarfsermittlung und Planung von Rekonfigurationen an Betriebsmitteln*: Herbert Utz Verlag, 2015.
- [58] X. L. Hoang, A. Fay, P. Marks, und M. Weyrich, "Systematization approach for the adaptation of manufacturing machines," in *2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA)*, 2016, S. 1–4.
- [59] M. M. Mabkhot, S. Darmoul, A. M. Al-Samhan, und A. Badwelan, "A Multi-Criteria Decision Framework Considering Different Levels of Decision-Maker Involvement to Reconfigure Manufacturing Systems," *Machines*, Bd. 8, Nr. 1, S. 8, 2020.
- [60] C. Bettaieb, A. J. Telmoudi, A. Sava, und L. Nabli, "Reconfigurable manufacturing system: Overview and proposition of new approach," in *2017 International Conference on Control, Automation and Diagnosis (IC-CAD)*, 2017, S. 534–539.
- [61] Y. Koren und M. Shpitalni, "Design of reconfigurable manufacturing systems," *Journal of Manufacturing Systems*, Bd. 29, Nr. 4, S. 130–141, 2010.
- [62] I. Maganha, A.-L. Andersen, C. Silva, und L. M. D. Ferreira, "A Classification of the Barriers in the Implementation Process of Reconfigurability," in *Towards Sustainable Customization: Bridging Smart Products and Manufacturing Systems: Proceedings of the 8th Changeable, Agile, Reconfigurable and Virtual Production Conference (CARV2021) and the 10th World Mass Customization & Personalization Conference (MCPC2021), Aalborg, Denmark, October/November 2021* 8, 2022, S. 88–95.
- [63] H. Haddou Benderbal, M. Dahane, und L. Benyoucef, "Modularity assessment in reconfigurable manufacturing system (RMS) design: an Archived Multi-Objective Simulated Annealing-based approach," *The international journal of advanced manufacturing technology*, Bd. 94, S. 729–749, 2018.
- [64] J. E. van Aken, "Management research based on the paradigm of the design sciences: the quest for field-tested and grounded technological rules," *Journal of management studies*, Bd. 41, Nr. 2, S. 219–246, 2004.
- [65] S. T. March und V. C. Storey, "Design science in the information systems discipline: an introduction to the special issue on design science research," *MIS quarterly*, S. 725–730, 2008.
- [66] A. Dresch, D. P. Lacerda, und J. A. V. Antunes, "Design science research," in *Design science research*: Springer, 2015, S. 67–102.
- [67] J. E. van Aken, "Management research as a design science: Articulating the research products of mode 2 knowledge production in management," *British journal of management*, Bd. 16, Nr. 1, S. 19–36, 2005.
- [68] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, und R. Lindgren, "Action design research," *MIS quarterly*, S. 37–56, 2011.
- [69] B. Kuechler und V. Vaishnavi, "Extending prior research with design science research: Two patterns for DSRIS project generation," in *International Conference on Design Science Research in Information Systems*, 2011, S. 166–175.
- [70] V. Çağdaş und E. Stubkjær, "Design research for cadastral systems," *Computers, Environment and Urban Systems*, Bd. 35, Nr. 1, S. 77–87, 2011.

- [71] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS quarterly*, S. 75–105, 2004.
- [72] A. Hevner and S. Chatterjee, “Design science research in information systems,” in *Design research in information systems*: Springer, 2010, S. 9–22.
- [73] T. Trepper, *Fundierung der Konstruktion agiler Methoden: Anpassung, Instanziierung und Evaluation der Methode PiK-AS*: Springer-Verlag, 2015.
- [74] T. Müller, B. Caesar, M. Weiß, S. Ferhat, N. Sahlab, and A. Fay *et al.*, “Reconfiguration management in manufacturing: A systematic literature review,” *at-Automatisierungstechnik*, Bd. 71, Nr. 5, S. 330–350, 2023.
- [75] T. Müller, S. Kamm, A. Löcklin, D. White, M. Mellinger, and N. Jazdi *et al.*, “Architecture and knowledge modelling for self-organized reconfiguration management of cyber-physical production systems,” *International Journal of Computer Integrated Manufacturing*, S. 1–22, 2022.
- [76] J. Wan, S. Tang, Di Li, M. Imran, C. Zhang, and C. Liu *et al.*, “Reconfigurable smart factory for drug packing in healthcare industry 4.0,” *IEEE Transactions on Industrial Informatics*, Bd. 15, Nr. 1, S. 507–516, 2018.
- [77] A. Bunte, P. Wunderlich, N. Moriz, P. Li, A. Mankowski, and A. Rogalla *et al.*, “Why symbolic ai is a key technology for self-adaption in the context of cpps,” in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2019, S. 1701–1704.
- [78] K. Balzereit und O. Niggemann, “AutoConf: A New Algorithm for Reconfiguration of Cyber-Physical Production Systems,” *IEEE Transactions on Industrial Informatics*, 2022.
- [79] M. Ashraf und F. Hasan, “Configuration selection for a reconfigurable manufacturing flow line involving part production with operation constraints,” *The international journal of advanced manufacturing technology*, Bd. 98, Nr. 5, S. 2137–2156, 2018.
- [80] M. Bortolini, F. G. Galizia, C. Mora, und F. Pilati, “Reconfigurability in cellular manufacturing systems: a design model and multi-scenario analysis,” *The international journal of advanced manufacturing technology*, Bd. 104, Nr. 9, S. 4387–4397, 2019.
- [81] A. S. Khan, L. Homri, J. Y. Dantan, und A. Siadat, “Modularity-based quality assessment of a disruptive reconfigurable manufacturing system-A hybrid meta-heuristic approach,” *The international journal of advanced manufacturing technology*, Bd. 115, Nr. 5, S. 1421–1444, 2021.
- [82] M. Ameer und M. Dahane, “Reconfigurability improvement in Industry 4.0: a hybrid genetic algorithm-based heuristic approach for a co-generation of setup and process plans in a reconfigurable environment,” *Journal of Intelligent manufacturing*, S. 1–23, 2021.
- [83] O. Battaia, A. Dolgui, N. Guschinsky, und F. Makssoud, “Mathematical Model for Processing Multiple Parts on Multi-positional Reconfigurable Machines with Turrets,” in *IFIP International Conference on Advances in Production Management Systems*, 2021, S. 563–573.
- [84] K. A. Kurniadi, S. Lee, und K. Ryu, “Digital twin approach for solving reconfiguration planning problems in rms,” in *IFIP International Conference on Advances in Production Management Systems*, 2018, S. 327–334.
- [85] A. Kumar, L. N. Pattanaik, und R. Agrawal, “Multi-objective Scheduling Model for Reconfigurable Assembly Systems,” in *Innovations in Soft Computing and Information Technology*: Springer, 2019, S. 209–217.
- [86] N. Rodrigues, E. Oliveira, und P. Leitão, “Decentralized and on-the-fly agent-based service reconfiguration in manufacturing systems,” *Computers in Industry*, Bd. 101, S. 81–90, 2018.
- [87] A. Ma, M. Frantzen, C. Snider, und A. Nassehi, “Anarchic manufacturing: Distributed control for product transition,” *Journal of Manufacturing Systems*, Bd. 56, S. 1–10, 2020.

- [88] N. Nikolakis, R. Senington, K. Sipsas, A. Syberfeldt, und S. Makris, "On a containerized approach for the dynamic planning and control of a cyber-physical production system," *Robotics and computer-integrated manufacturing*, Bd. 64, S. 101919, 2020.
- [89] M. Mahmoodjanloo, R. Tavakkoli-Moghaddam, A. Baboli, und A. Bozorgi-Amiri, "Flexible job shop scheduling problem with reconfigurable machine tools: An improved differential evolution algorithm," *Applied Soft Computing*, Bd. 94, S. 106416, 2020.
- [90] M. Krä, L. Vogt, C. Härdtlein, S. Schiele, und J. Schilp, "Production planning for collaborating resources in cyber-physical production systems," *Procedia CIRP*, Bd. 93, S. 192–197, 2020.
- [91] A. Villalonga, E. Negri, L. Fumagalli, M. Macchi, F. Castaño, und R. Haber, "Local decision making based on distributed digital twin framework," *IFAC-PapersOnLine*, Bd. 53, Nr. 2, S. 10568–10573, 2020.
- [92] T. B. Sheridan und W. L. Verplank, "Human and computer control of undersea teleoperators," Massachusetts Inst of Tech Cambridge Man-Machine Systems Lab, 1978.
- [93] R. C. Martin, "Design principles and design patterns," *Object Mentor*, Bd. 1, Nr. 34, S. 597, 2000.
- [94] P. Leitão, J. Barbosa, A. Pereira, J. Barata, und A. W. Colombo, "Specification of the PERFoRM architecture for the seamless production system reconfiguration," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, S. 5729–5734.
- [95] J.-P. Schmidt, T. Müller, und M. Weyrich, "Einsatz einer service-orientierten Architektur zur Orchestrierung eines dezentralen Intralogistiksystems," 2020.
- [96] J. O. Kephart und D. M. Chess, "The vision of autonomic computing," *Computer*, Bd. 36, Nr. 1, S. 41–50, 2003.
- [97] A. Musil, J. Musil, D. Weyns, T. Bures, H. Muccini, und M. Sharaf, "Patterns for self-adaptation in cyber-physical systems," in *Multi-disciplinary engineering for cyber-physical production systems*: Springer, 2017, S. 331–368.
- [98] M. Hankel und B. Rexroth, "The reference architectural model industrie 4.0 (rami 4.0)," *ZVEI*, Bd. 2, Nr. 2, S. 4–9, 2015.
- [99] S. K. Panda, T. Schröder, L. Wisniewski, und C. Diedrich, "Plug&Produce Integration of Components into OPC UA based data-space," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2018, Bd. 1, S. 1095–1100.
- [100] J. Arm, T. Benes, P. Marcon, Z. Bradac, T. Schröder, und A. Belyaev *et al.*, "Automated design and integration of asset administration shells in components of industry 4.0," *Sensors*, Bd. 21, Nr. 6, S. 2004, 2021.
- [101] A. Hennecke und M. Ruskowski, "Design of a flexible robot cell demonstrator based on CPPS concepts and technologies," in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, 2018, S. 534–539.
- [102] M. Onori, N. Lohse, J. Barata, und C. Hanisch, "The IDEAS project: plug & produce at shop-floor level," *Assembly automation*, 2012.
- [103] A. W. Colombo, S. Karnouskos, und others, "Towards the factory of the future: A service-oriented cross-layer infrastructure," *ICT shaping the world: a scientific view*, Bd. 65, S. 81, 2009.
- [104] P. Leitão, N. Rodrigues, C. Turrin, und A. Pagani, "Multiagent system integrating process and quality control in a factory producing laundry washing machines," *IEEE Transactions on Industrial Informatics*, Bd. 11, Nr. 4, S. 879–886, 2015.
- [105] A. W. Colombo, T. Bangemann, und S. Karnouskos, "IMC-AESOP outcomes: Paving the way to collaborative manufacturing systems," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, S. 255–260.

- [106] P. Leitão, J. Barbosa, A. Pereira, J. Barata, and A. W. Colombo, "Specification of the PERFoRM architecture for the seamless production system reconfiguration," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, 2016, S. 5729–5734.
- [107] J. Lee, B. Bagheri, and H.-A. Kao, "A cyber-physical systems architecture for industry 4.0-based manufacturing systems," *Manufacturing letters*, Bd. 3, S. 18–23, 2015.
- [108] M. Weyrich und C. Ebert, "Reference architectures for the internet of things," *IEEE Software*, Bd. 33, Nr. 1, S. 112–116, 2015.
- [109] M. Hoffmann, T. Meisen, D. Schilberg, und S. Jeschke, "Multi-dimensional production planning using a vertical data integration approach: A contribution to modular factory design," in *2013 10th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT)*, 2013, S. 1–6.
- [110] S. Schmied, D. Großmann, R. K. Mueller, S. G. Mathias, und U. Jumar, "Erstellung und Management von Informationsmodellen für bestehende Produktionssysteme," *at-Automatisierungstechnik*, Bd. 68, Nr. 5, S. 325–336, 2020.
- [111] C. Hildebrandt, A. Köcher, C. Küstner, C.-M. López-Enríquez, A. W. Müller, und B. Caesar *et al.*, "Ontology building for cyber-physical systems: Application in the manufacturing domain," *IEEE Transactions on Automation Science and Engineering*, Bd. 17, Nr. 3, S. 1266–1282, 2020.
- [112] V. Jirkovský, Marek Obitko, und Vladimír Mařík, "Understanding data heterogeneity in the context of cyber-physical systems integration," *IEEE Transactions on Industrial Informatics*, Bd. 13, Nr. 2, S. 660–667, 2016.
- [113] S. Kamm, N. Jazdi, und M. Weyrich, "Knowledge Discovery in Heterogeneous and Unstructured Data of Industry 4.0 Systems: Challenges and Approaches," *Procedia CIRP*, Bd. 104, S. 975–980, 2021.
- [114] R. Herzog, M. Jacoby, und I. P. Žarko, "Semantic interoperability in IoT-based automation infrastructures," *at-Automatisierungstechnik*, Bd. 64, Nr. 9, S. 742–749, 2016.
- [115] A. Köcher, C. Hildebrandt, da Silva, Luis Miguel Vieira, und A. Fay, "A formal capability and skill model for use in plug and produce scenarios," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020, Bd. 1, S. 1663–1670.
- [116] C. Diedrich, A. Belyaev, T. Schröder, J. Vialkowitsch, A. Willmann, und T. Usländer *et al.*, "Semantic interoperability for asset communication within smart factories," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, S. 1–8.
- [117] D. Mourtzis, N. Milas, und N. Athinaios, "Towards machine shop 4.0: a general machine model for CNC machine-tools through OPC-UA," *Procedia CIRP*, Bd. 78, S. 301–306, 2018.
- [118] R. Beregi, G. Pedone, B. Házy, und J. Váncza, "Manufacturing execution system integration through the standardization of a common service model for cyber-physical production systems," *Applied Sciences*, Bd. 11, Nr. 16, S. 7581, 2021.
- [119] H. K. Pakala, A. Belyaev, und C. Diedrich, "Middleware architecture for application layer interoperability of standardized digital representations," in *IECON 2021-47th Annual Conference of the IEEE Industrial Electronics Society*, 2021, S. 1–6.
- [120] Vogel-Heuser B, Fay A, Seitz M, Gehlhoff F, "Agenten zur Realisierung von Industrie 4.0," *VDI/VDE-Gesellschaft Mess-und Automatisierungstechnik*, 2019.
- [121] C. Feilmayr und W. WöB, "An analysis of ontologies and their success factors for application to business," *Data & Knowledge Engineering*, Bd. 101, S. 1–23, 2016.



- [122]N. Sahlab, S. Kamm, T. Müller, N. Jazdi, und M. Weyrich, "Knowledge graphs as enhancers of intelligent digital twins," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, S. 19–24.
- [123]R. Studer, V. R. Benjamins, und D. Fensel, "Knowledge engineering: principles and methods," *Data & Knowledge Engineering*, Bd. 25, 1-2, S. 161–197, 1998.
- [124]A. Hengstebeck, A. Barthelmey, und J. Deuse, "Reconfiguration assistance for cyber-physical production systems," in *Tagungsband des 3. Kongresses Montage Handhabung Industrieroboter*: Springer, 2018, S. 177–186.
- [125]C. Siedelhofer, J. Schallow, P. Wolf, S. Mayer, und J. Deuse, "Simulationsbasierte Rekonfigurationsplanung flexibler Montagesysteme," *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Bd. 113, Nr. 4, S. 216–219, 2018.
- [126]X.-L. Hoang, S. Backhaus, R. Bense, A. Fay, D. Küstner, und B. Schulze, "An interface-oriented resource capability model to support reconfiguration of manufacturing systems," in *2019 IEEE International Systems Conference (SysCon)*, 2019, S. 1–8.
- [127]Verein Deutscher Ingenieure, "VDI/VDE 3682-1, Formalisierte Prozessbeschreibungen - Konzept und grafische Darstellung," 2015.
- [128]F. Ocker, B. Vogel-Heuser, und C. J. J. Paredis, "Applying semantic web technologies to provide feasibility feedback in early design phases," *Journal of Computing and Information Science in Engineering*, Bd. 19, Nr. 4, 2019.
- [129]E. Järvenpää, N. Siltala, O. Hylli, und M. Lanz, "The development of an ontology for describing the capabilities of manufacturing resources," *Journal of Intelligent manufacturing*, Bd. 30, Nr. 2, S. 959–978, 2019.
- [130]E. Järvenpää, N. Siltala, O. Hylli, und M. Lanz, "Product Model ontology and its use in capability-based matchmaking," *Procedia CIRP*, Bd. 72, S. 1094–1099, 2018.
- [131]B. Maschler, T. Müller, A. Löcklin, und M. Weyrich, "Transfer Learning as an Enhancement for Reconfiguration Management of Cyber-Physical Production Systems," *Procedia CIRP*, Bd. 112, S. 220–225, 2022.
- [132]B. Caesar, D. M. Tilbury, K. Barton, und A. Fay, "Requirements for Reconfiguration Management for Manufacturing Systems," *IFAC-PapersOnLine*, Bd. 55, Nr. 10, S. 3214–3219, 2022.
- [133]A. Azab, H. ElMaraghy, P. Nyhuis, J. Pachow-Frauenhofer, und M. Schmidt, "Mechanics of change: A framework to reconfigure manufacturing systems," *CIRP Journal of Manufacturing Science and Technology*, Bd. 6, Nr. 2, S. 110–119, 2013.
- [134]M. Bortolini, F. G. Galizia, und C. Mora, "Reconfigurable manufacturing systems: Literature review and research trend," *Journal of Manufacturing Systems*, Bd. 49, S. 93–106, 2018.
- [135]F. Karl, G. Reinhart, und M. F. Zaeh, "Strategic planning of reconfigurations on manufacturing resources," *Procedia CIRP*, Bd. 3, S. 608–613, 2012.
- [136]J. J. Downs und E. F. Vogel, "A plant-wide industrial process control problem," *Computers & chemical engineering*, Bd. 17, Nr. 3, S. 245–255, 1993.
- [137]R. G. Landers, B.-K. Min, und Y. Koren, "Reconfigurable machine tools," *CIRP annals*, Bd. 50, Nr. 1, S. 269–274, 2001.
- [138]G. Hoellthaler, M. Schreiber, K. Vernickel, J. Be Isa, J. Fischer, und N. Weinert *et al.*, "Reconfiguration of production systems using optimization and material flow simulation," *Procedia CIRP*, Bd. 81, S. 133–138, 2019.
- [139]K. K. Goyal, P. K. Jain, und M. Jain, "Optimal configuration selection for reconfigurable manufacturing system using NSGA II and TOPSIS," *International Journal of Production Research*, Bd. 50, Nr. 15, S. 4175–4191, 2012.

- [140]S. K. Moghaddam, M. Houshmand, K. Saitou, and O. Fatahi Valilai, "Configuration design of scalable reconfigurable manufacturing systems for part family," *International Journal of Production Research*, Bd. 58, Nr. 10, S. 2974–2996, 2020.
- [141]F. A. Touzout and L. Benyoucef, "Multi-objective multi-unit process plan generation in a reconfigurable manufacturing environment: a comparative study of three hybrid metaheuristics," *International Journal of Production Research*, Bd. 57, Nr. 24, S. 7520–7535, 2019.
- [142]P. Marks, *Softwaregestütztes Assistenzkonzept zur Modernisierung von Automatisierungssystemen*: Shaker, 2020.
- [143]P. Dayan and C. Watkins, "Q-learning," *Machine learning*, Bd. 8, Nr. 3, S. 279–292, 1992.
- [144]X.-L. Hoang, C. Hildebrandt, and A. Fay, "Product-oriented description of manufacturing resource skills," *IFAC-PapersOnLine*, Bd. 51, Nr. 11, S. 90–95, 2018.
- [145]M. Krä, L. Vogt, C. Härdtlein, S. Schiele, and J. Schilp, "Production planning for collaborating resources in cyber-physical production systems," *Procedia CIRP*, Bd. 93, S. 192–197, 2020.
- [146]C. El Zant, K. Benfriha, S. Loubère, A. Aoussat, and O. Adjoul, "A design methodology for modular processes orchestration," *CIRP Journal of Manufacturing Science and Technology*, Bd. 35, S. 106–117, 2021.
- [147]B. Caesar, M. Nieke, A. Köcher, C. Hildebrandt, C. Seidl, and A. Fay *et al.*, "Context-sensitive reconfiguration of collaborative manufacturing systems," *IFAC-PapersOnLine*, Bd. 52, Nr. 13, S. 307–312, 2019.
- [148]I. S. Khan, U. Ghafoor, and T. Zahid, "Meta-heuristic approach for the development of alternative process plans in a reconfigurable production environment," *IEEE Access*, Bd. 9, S. 113508–113520, 2021.
- [149]E. Asghar, U. K. U. Zaman, A. A. Baqai, and L. Homri, "Optimum machine capabilities for reconfigurable manufacturing systems," *The international journal of advanced manufacturing technology*, Bd. 95, S. 4397–4417, 2018.
- [150]H. Zhu, M. Chen, Z. Zhang, and D. Tang, "An adaptive real-time scheduling method for flexible job shop scheduling problem with combined processing constraint," *IEEE Access*, Bd. 7, S. 125113–125121, 2019.
- [151]B. Kaiser, A. Reichle, and A. Verl, "Model-based automatic generation of digital twin models for the simulation of reconfigurable manufacturing systems for timber construction," *Procedia CIRP*, Bd. 107, S. 387–392, 2022.
- [152]S. Sierla, V. Kyrki, P. Aarnio, and V. Vyatkin, "Automatic assembly planning based on digital product descriptions," *Computers in Industry*, Bd. 97, S. 34–46, 2018.
- [153]Z. Guo, Y. Zhang, X. Zhao, and X. Song, "CPS-based self-adaptive collaborative control for smart production-logistics systems," *IEEE transactions on cybernetics*, Bd. 51, Nr. 1, S. 188–198, 2020.
- [154]T. Müller, B. Maschler, D. Dittler, N. Jazdi, and M. Weyrich, "Intelligent Exploration of Solution Spaces Exemplified by Industrial Reconfiguration Management," *arXiv preprint arXiv:2207.01693*, 2022.
- [155]V. D.I. VDI, "3633 Blatt 12-Simulation of systems in materials handling, logistics and production-Simulation and optimisation," *Accessed on: Nov*, Bd. 26, 2020.
- [156]A. M.S. FIPA, "Fipa agent management specification," *Standard component SC00023K, Foundation for Intelligent Physical Agents*, Bd. 2, 2004.
- [157]K. Amine, "Multiobjective simulated annealing: Principles and algorithm variants," *Advances in Operations Research*, Bd. 2019, 2019.
- [158]R. V. Rao, *Decision making in the manufacturing environment: using graph theory and fuzzy multiple attribute decision making methods*: Springer, 2007.
- [159]J.-P. Schmidt, T. Müller, and M. Weyrich, "Methodology for the model driven development of service oriented plant controls," *Procedia CIRP*, Bd. 67, S. 173–178, 2018.

## Anhang A: Konzeptdetails

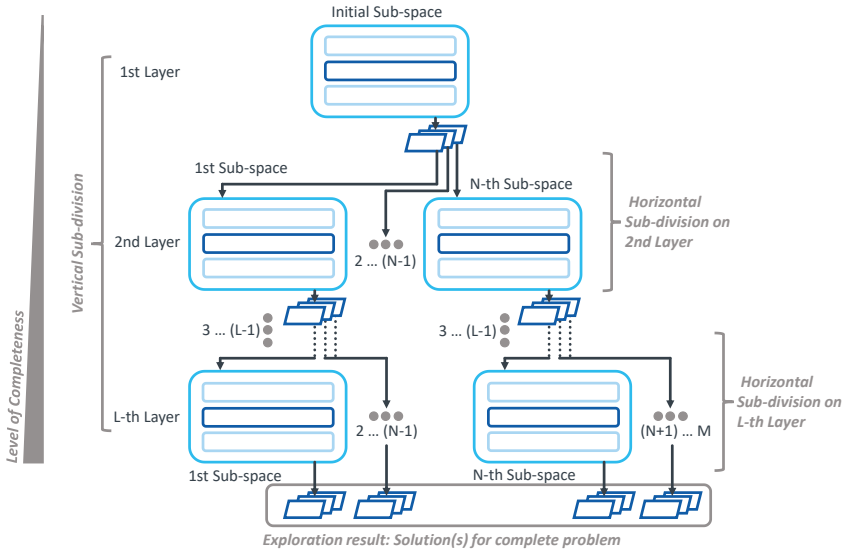


Abbildung A.1: Universelle Methodik für die intelligente Exploration des Lösungsraums [154]

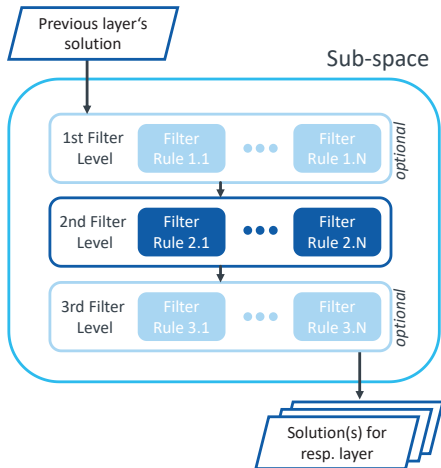
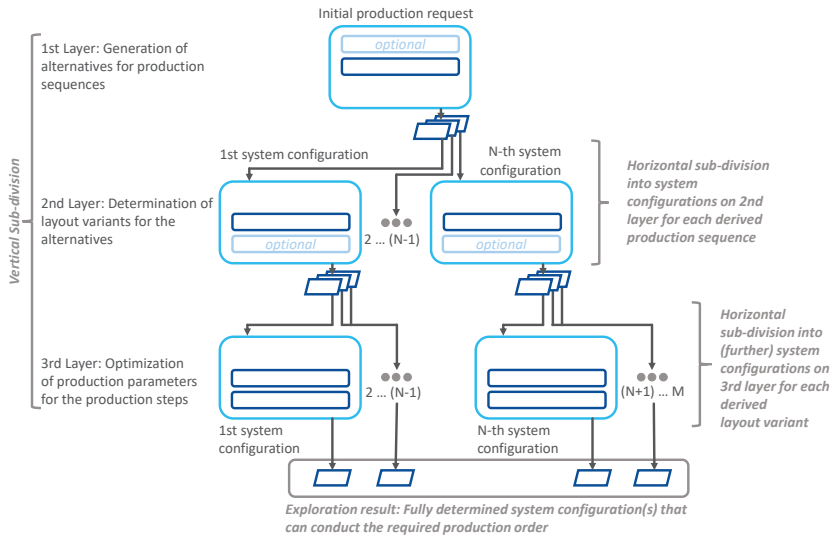
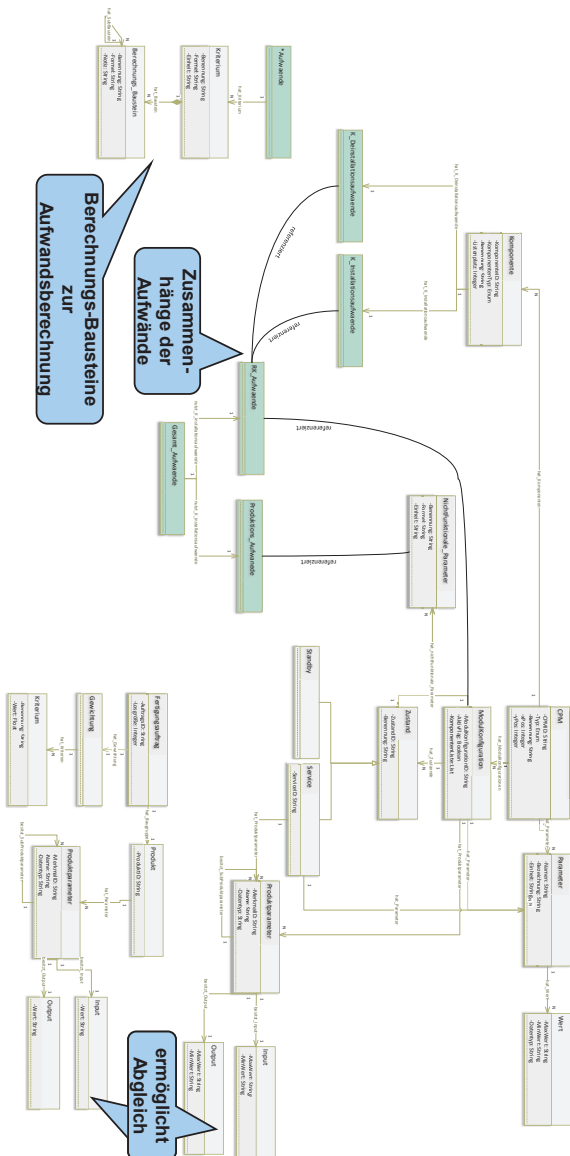


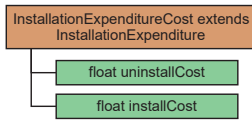
Abbildung A.2: Filterstufen innerhalb jedes Teil-Lösungsraums [154]



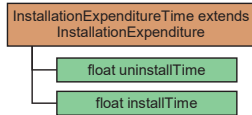
**Abbildung A.3: Intelligente Exploration des Lösungsraums für den Anwendungsfall des selbstorganisierten Rekonfigurationsmanagements [154]**



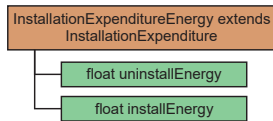
**Abbildung A.4: Detaillierter Überblick der Informationsmodellierung**



**Abbildung A.5: Installations- und Deinstallationskosten eines Moduls im Produktionslayout**



**Abbildung A.6: Installations- und Deinstallationszeit eines Moduls im Produktionslayout**



**Abbildung A.7: Installations- und Deinstallationsenergie eines Moduls im Produktionslayout**

## Anhang B: Realisierungsdetails

**Tabelle B.1:** Gewählte Parameter für die Realisierung der GA-basierten Verfahrensvarianten in Abhängigkeit der Anzahl von eingesetzten CPPMs

| Gemeinsame Parameter aller GA-Verfahrensvarianten                         |          |          |          |          |          |
|---------------------------------------------------------------------------|----------|----------|----------|----------|----------|
| Anzahl CPPM                                                               | 2        | 3        | 4        | 5        | default  |
| Populationsgröße                                                          | 10       | 20       | 26       | 30       | 20       |
| Anzahl an Generationen                                                    | 100      | 200      | 300      | 500      | 250      |
| Selbstadaptive Kreuzungswahrscheinlichkeit                                | 0,8–0,5  | 0,8–0,5  | 0,9–0,6  | 0,9–0,6  | 0,9–0,6  |
| Mutationswahrscheinlichkeit                                               | 0,2-0,05 | 0,2-0,05 | 0,2-0,05 | 0,2-0,05 | 0,2-0,05 |
| Zusätzliche-Parameter der rekfigurationsoptimierten GA-Verfahrensvariante |          |          |          |          |          |
| Teilnehmeranzahl der Wettkampf Selektion                                  | 3 - 9    | 5 - 11   | 7 - 13   | 8 - 14   | 5 - 11   |

**Tabelle B.2:** Rekonfigurationsmatrix: zeitliche Rekonfigurationsaufwände beim Überführen in die verschiedenen Modulkonfigurationen des Bohrmoduls

|   | A         | B         | C         |
|---|-----------|-----------|-----------|
| A | 0 (A=>A)  | 10 (B=>A) | 20 (C=>A) |
| B | 10 (A=>B) | 0 (B=>B)  | 20 (C=>B) |
| C | 20 (A=>C) | 20 (B=>C) | 0 (C=>C)  |

```

graph TD
    N2_CPTM1((N2_CPTM1)) -- "(nodeNr=2)" --> Ready
    subgraph Ready_Box [Ready]
        direction TB
        Ready_en["en:  
cpmReady(nodeNr)=1;  
ex:  
cpmReady(nodeNr)=0;"]
    end
    Ready_Box --> transport1
    subgraph transport1_Box [transport1]
        direction TB
        transport1_en["en:  
PcurrPos(i)=nodeNr;  
cpmReserved(nodeNr)=0;  
serviceCountCPTM(1,1)=serviceCountCPTM(1,1)+1;"]
    end
    transport1_Box --> Join(( ))
    Join --> Wait(( ))
    Wait --> Final(( ))
    Join -- "after(tServiceCPTM(1,1),sec) ...  
([PnextPos(i),PobjPos(i),cpmReserved]=nextService(i,PcurrPos(i), ...  
PobjService(i),cpmReady,cpmReserved,tServiceCPTM))" --> Ready
    Join -- "[PnextPos(i)~=0 && cpmReady(PnextPos(i))=1 ...  
&& (cpmReserved(PnextPos(i))=0 || cpmReserved(PnextPos(i))=1) ...  
(cpmReserved(PnextPos(i))=1,cpmReserved(PobjPos(i)) ...  
=1,P.data=[Pid(i) PnextPos(i) PobjPos(i)],send(P))" --> transport1
    Wait -- "Wait  
ex:  
[PnextPos(i),PobjPos(i),cpmReserved]=nextService(i,PcurrPos(i), ...  
PobjService(i),cpmReady,cpmReserved,tServiceCPTM);  
after(1,sec)" --> transport1

```

**Abbildung B.2: Subchart: CPTM-Block**



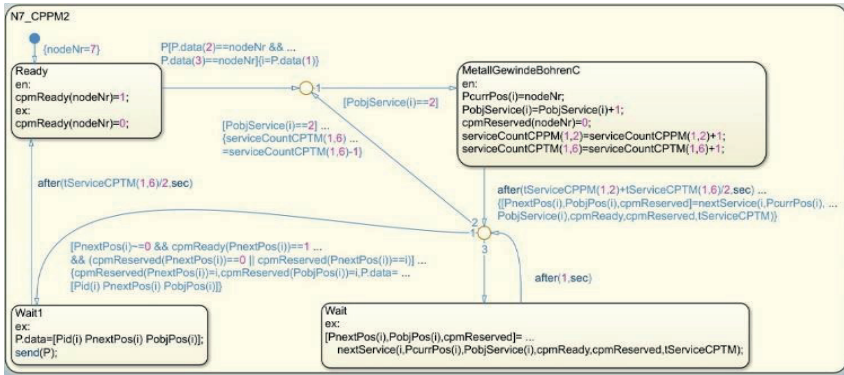


Abbildung B.3: Subchart: CPPM-Block

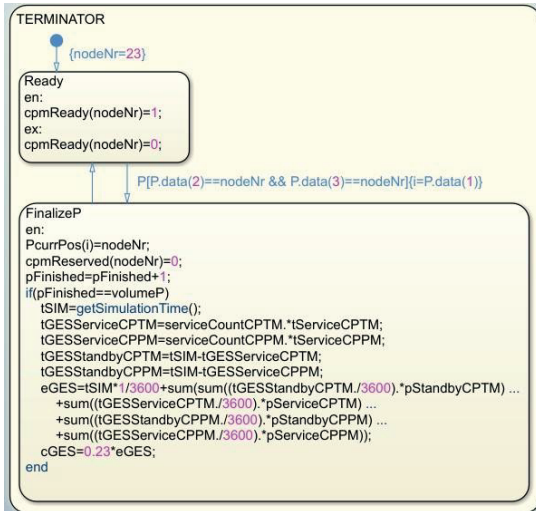


Abbildung B.4: Subchart: Terminator-Block

Um am Beispiel aufzuzeigen wie der Energieaufwand konkret berechnet wird, werden nachfolgend Teile des Bohrmoduls, das im Zuge der Evaluierung genutzt wurde, beschrieben um einen tiefer gehenden Einblick zu liefern. Auf eine vollständige Beschreibung eines Produktionsmoduls wird aufgrund des enormen Umfangs jedoch verzichtet, da beispielsweise das Bohrmodul über fünf Modulkonfigurationen mit ihren jeweiligen Services verfügt und insbesondere deren merkmalsbasierte Beschreibung zu einer XML-Datei mit 2194 Zeilen führt. Im Allgemeinen basiert die Beschreibung der Module und ihrer Konfigurationsalternativen und Services auf Informatio-

nen aus der Literatur (insbesondere Normen wie der DIN 8580 zu Fertigungsverfahren) und Datenblättern (z.B. des Werkzeugmaschinenherstellers TRUMPF SE + Co. KG), wobei auch Annahmen getroffen wurden.

Mit

$$p_{service} = 1,5 * p_{standby} + p_{antrieb} \quad (28)$$

ergibt sich für das Bohrmodul und dessen Leistungsaufnahme bei der Ausführung eines Services zum Bohren von Löchern unter Verwendung des Vorschub pro Umdrehung  $f_n$ , des Schneiddurchmessers  $D_c$ , der Drehzahl  $n$  mit einem ergänzten Exponenten  $q$  zur Abbildung von Reibungsverlusten, der spezifischen Schnittkraft  $k_c$  und dem Wirkungsgrad  $\eta$  :

$$p_{service} = 1,5 * p_{standby} + \frac{f_n D_c^2 \pi k_c}{240 * 10^6 \eta} n^q = 1,5 * p_{standby} + c_1 n^q \quad (29)$$

Basierend auf der Drehzahl  $n$ , der Bohrtiefe  $l_m$  und dem Vorschub pro Umdrehung  $f_n$  ergibt sich  $t_{service}$  zu:

$$t_{service} = \frac{60 * l_m}{f_n * n} = \frac{c_2}{n} \quad (30)$$

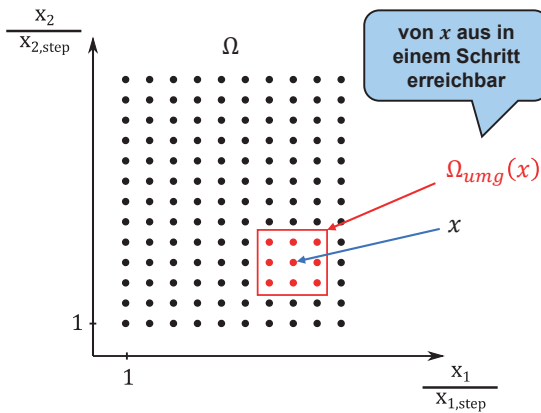
Daraus ergibt sich der Service-Energieaufwand  $e_{service}$ :

$$e_{service} = p_{service} * t_{service} = \frac{1,5 * p_{standby}}{n} c_2 + c_1 c_2 n^{q-1} \quad (31)$$

Für die eigentliche Berechnung muss die Drehzahl als zu optimierender Produktionsparameter festgelegt werden, hierzu sind beispielsweise für den Service *MetallLochBohrenA* der Modulkonfiguration *BohrenA* Drehzahlen im Bereich von 800 – 1600 mit einer Schrittweite von 200 einstellbar. Die Bohrtiefe  $l_m$  (auch Bohrloch-Tiefe oder *depth*) wird dem jeweiligen Produktionsauftrag entnommen. Der Wirkungsgrad  $\eta$  wurde auf 95% gesetzt,  $k_c$  beträgt für Metall 355.655,88 und  $q=1,35$  wurde gewählt. Der Schneiddurchmesser  $D_c$  beträgt 10 mm, der Vorschub pro Umdrehung  $f_n$  für Metall beträgt 0,1 mm und die Leistungsaufnahme im Standby  $p_{standby}$  ist 2 kW.

| Name                  | Value                               |
|-----------------------|-------------------------------------|
| CPPM                  | 1x1 struct                          |
| CPTM                  | 1x1 struct                          |
| cycleTimeP            | 5000                                |
| decisionMaking        | 0                                   |
| DVValueCPPM_max       | [1600,2400]                         |
| DVValueCPPM_min       | [800,1200]                          |
| DVValueCPPM_step      | [200,200]                           |
| DVValueCPTM_max       | 1x21 int32                          |
| DVValueCPTM_min       | 1x21 int32                          |
| initDVValueCPPM       | [1200,1800]                         |
| initDVValueCPTM       | 1x21 int32                          |
| iterations            | 100                                 |
| OptimizationAlgorithm | "MultiobjectiveSA"                  |
| positionCPPM          | [3,7,15]                            |
| s                     | 1x26 int32                          |
| serviceOrder          | 1x3 string                          |
| simFolder             | "Simulation_28_04_2022_06_09_59"    |
| systemName            | "AgentSystemconfiguration0_2_6_0_0" |
| t                     | 1x26 int32                          |
| volumeP               | 3                                   |
| weightCost            | 0.2500                              |
| weightEnergy          | 0.2500                              |
| weightTime            | 0.5000                              |

Abbildung B.5: Beispiel des initialisierten MATLAB Workspace einer Systemkonfiguration

Abbildung B.6: Zweidimensionale Visualisierung der Umgebung von  $x \in \Omega \subset \mathbb{R}^2$

In Abbildung B.7 ist der Einfluss von  $\alpha$  auf die Akzeptanzwahrscheinlichkeit  $P(y)$  im Rahmen des in Kapitel 4.1 vorgestellten MOSA visualisiert, dabei gilt hier  $w_1 = w_2 = 0.5$  und  $c = 1/5$ . Zudem ist zu beachten, dass sich der akutelle Lösungspunkt  $F(x)$  jeweils im Nullpunkt befindet.

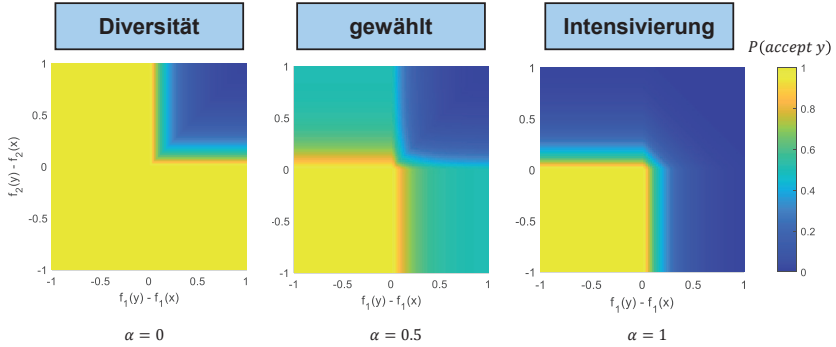


Abbildung B.7: Zweidimensionale Visualisierung des Einflusses von  $\alpha$  auf die Akzeptanzwahrscheinlichkeit

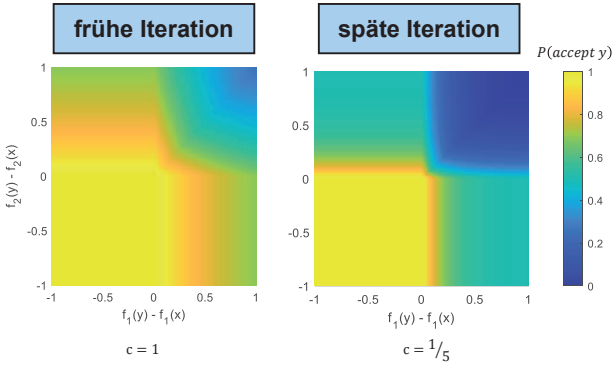


Abbildung B.8: Zweidimensionale Visualisierung des Einflusses der Temperatur  $c$  auf die Akzeptanzwahrscheinlichkeit

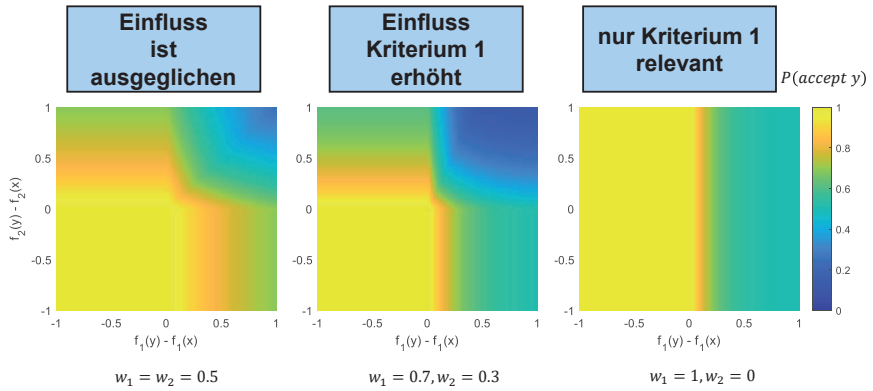


Abbildung B.9: Zweidimensionale Visualisierung des Einflusses der Gewichtung  $w_i$  auf die Akzeptanzwahrscheinlichkeit

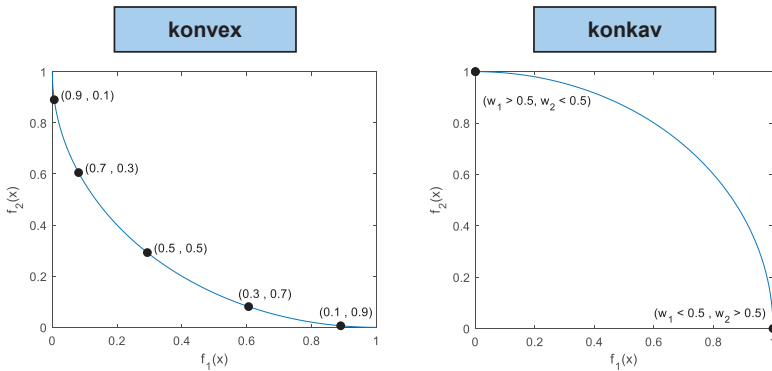


Abbildung B.10: Zweidimensionale Beispiele für die Auswahl von  $\tilde{x}$  aus einer Pareto-Front für unterschiedliche Gewichtungen ( $w_1, w_2$ ) basierend auf der Methode der gewichteten Summe

In Abbildung B.10 ist ersichtlich, dass für eine konvexe Pareto-Front unterschiedliche Lösungen in Abhängigkeit der Gewichtungen gefunden werden, wohingegen für eine konkave Pareto-Front nur zwei Punkte für eine Vielzahl unterschiedlicher Gewichtungen gefunden werden.

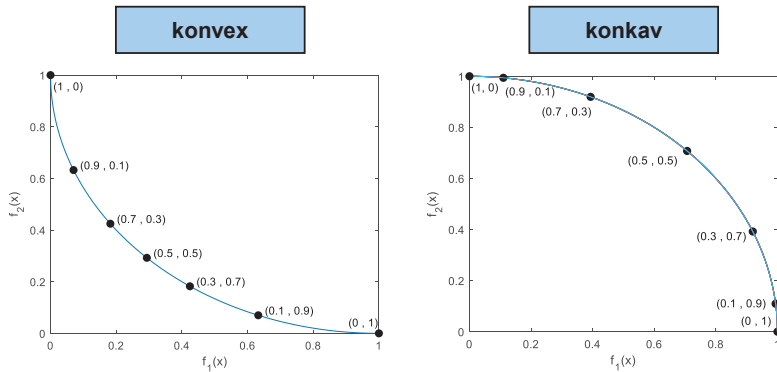


Abbildung B.11: Zweidimensionale Beispiele für die Auswahl von  $\tilde{x}$  aus einer Pareto-Front für unterschiedliche Gewichtungen  $(w_1, w_2)$  basierend auf dem Tehebycheff-Problem

Der Vergleich von Abbildung B.10 und Abbildung B.11 verdeutlicht die Auswahl unterschiedlicher Kompromisslösungen  $\tilde{x}^*$  bei gleichbleibender Gewichtung.

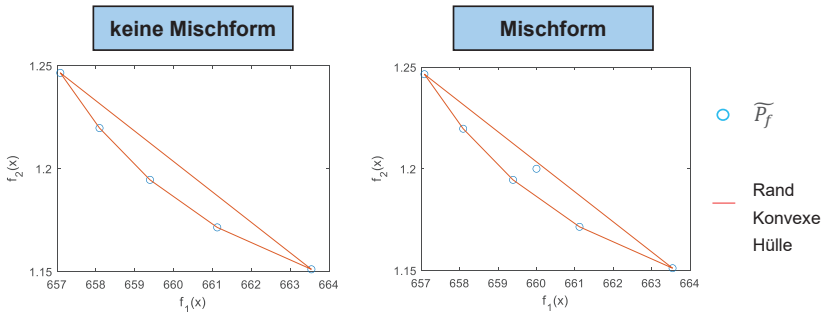


Abbildung B.12: Zwei Beispiele für eine konvexe Hülle um die Punktemenge einer Pareto-Front im zweidimensionalen Raum

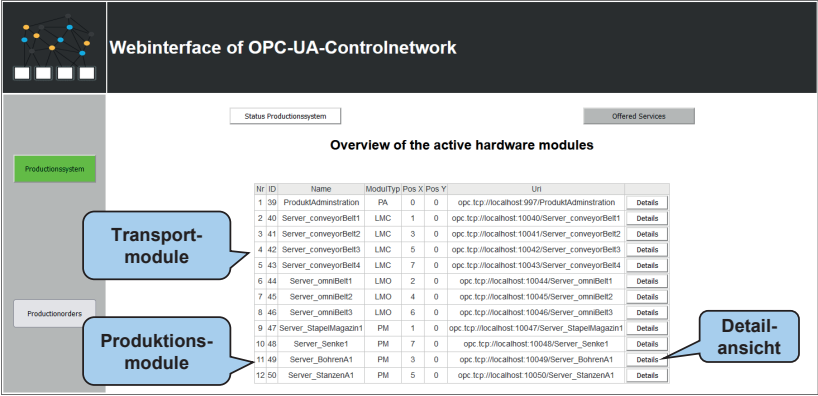


Abbildung B.13: Webinterface der SBC – Übersicht der Module im Steuerungsnetzwerk

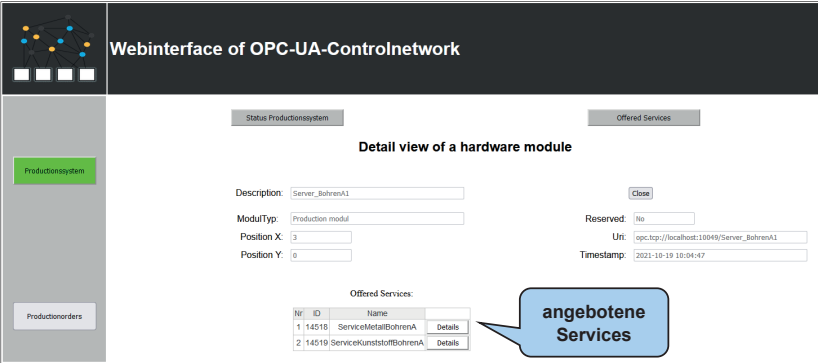


Abbildung B.14: Webinterface der SBC – Detailansicht eines Produktionsmoduls im Steuerungsnetzwerk

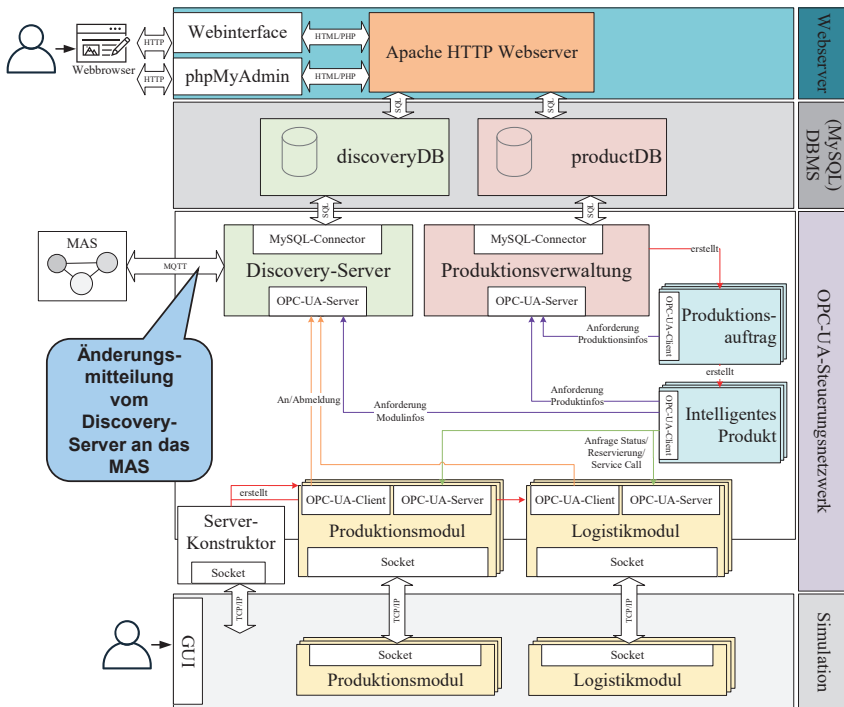


Abbildung B.15: Detaillierte Übersicht der Umsetzung des Control- und des Asset-Layers mit Anbindung des zugehörigen Webinterface zur Eingabe von Produktionsaufträgen und Überwachung des Produktionssystems

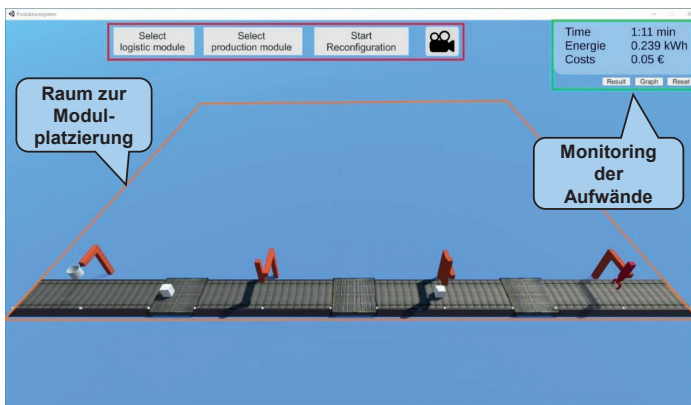


Abbildung B.16: Unity-GUI: Übersicht der Hauptansicht



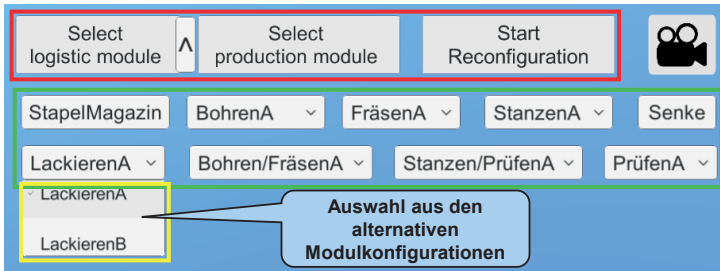


Abbildung B.17: Unity-GUI: Auswahl von Modulkonfigurationen welche per Drag & Drop ins Produktions-layout integriert werden können

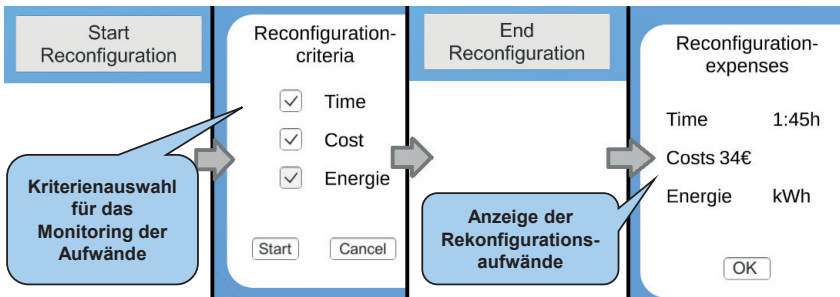


Abbildung B.18: Unity-GUI: Übersicht der Elemente für das Monitoring der Rekonfigurationsaufwände

## Anhang C: Evaluierungsdetails

Die Rekonfigurationsaufwände auf Systemebene berechnen sich basierend auf der Festlegung von 30 Minuten Rekonfigurationszeit für das Entfernen eines Produktionsmoduls aus dem Produktions-Layout und 60 Minuten für das Hinzufügen eines Produktionsmoduls. Für die Rekonfigurationsaufwände auf Maschinen- bzw. Modulebene gilt die nachfolgende Tabelle.

**Tabelle C.1: Installations- und Deinstallationsaufwände für die Komponenten der Produktionsmodule**

| Komponententyp       | Komponenten-ID | Installationszeit | Deinstallationszeit |
|----------------------|----------------|-------------------|---------------------|
| <b>Modul Bohren</b>  |                |                   |                     |
| <b>Hardware</b>      | 1              | 5                 | 5                   |
|                      | 2              | 5                 | 5                   |
|                      | 3              | 5                 | 5                   |
|                      | 4              | 20                | 10                  |
|                      | 5              | 20                | 10                  |
| <b>Software</b>      | 6              | 10                | 0                   |
|                      | 7              | 10                | 0                   |
| <b>Modul Fräsen</b>  |                |                   |                     |
| <b>Hardware</b>      | 1              | 10                | 10                  |
|                      | 2              | 5                 | 5                   |
|                      | 3              | 20                | 10                  |
|                      | 4              | 20                | 10                  |
| <b>Software</b>      | 5              | 10                | 0                   |
|                      | 6              | 10                | 0                   |
| <b>Modul Stanzen</b> |                |                   |                     |
| <b>Hardware</b>      | 1              | 10                | 5                   |
|                      | 2              | 5                 | 5                   |
| <b>Software</b>      | 3              | 10                | 0                   |
| <b>Modul Prüfen</b>  |                |                   |                     |
| <b>Hardware</b>      | 1              | 10                | 10                  |
|                      | 2              | 10                | 5                   |
| <b>Software</b>      | 3              | 10                | 0                   |

| Komponententyp              | Komponenten-ID | Installationszeit | Deinstallationszeit |
|-----------------------------|----------------|-------------------|---------------------|
|                             | 4              | 10                | 0                   |
| <b>Modul Lackieren</b>      |                |                   |                     |
| <b>Hardware</b>             | 1              | 10                | 5                   |
|                             | 2              | 10                | 5                   |
| <b>Software</b>             | 3              | 10                | 0                   |
| <b>Modul Bohren/Fräsen</b>  |                |                   |                     |
| <b>Hardware</b>             | 1              | 5                 | 5                   |
|                             | 2              | 5                 | 5                   |
|                             | 3              | 20                | 10                  |
|                             | 4              | 20                | 10                  |
|                             | 5              | 5                 | 5                   |
| <b>Software</b>             | 6              | 10                | 0                   |
|                             | 7              | 10                | 0                   |
| <b>Modul Stanzen/Prüfen</b> |                |                   |                     |
| <b>Hardware</b>             | 1              | 10                | 5                   |
|                             | 2              | 5                 | 5                   |
|                             | 3              | 10                | 5                   |
|                             | 4              | 5                 | 5                   |
| <b>Software</b>             | 5              | 10                | 0                   |
|                             | 6              | 10                | 0                   |
|                             | 7              | 10                | 0                   |

Zur Bestimmung der Rekonfigurationskosten auf der Maschinen- und Systemebene werden die Personalkosten eines Werkarbeiters mit 15 €/h angesetzt und entsprechend mit den zeitlichen Aufwänden multipliziert.

| Name         | MerkmalID | Input | Output |
|--------------|-----------|-------|--------|
| Loch         | 100       | False | True   |
| x-Koordinate | 110       | -     | 35     |
| y-Koordinate | 120       | -     | 35     |
| z-Koordinate | 130       | -     | 8      |
| Durchmesser  | 140       | -     | 10     |
| Tiefe        | 150       | -     | 8      |
| Loch         | 100       | False | True   |
| x-Koordinate | 110       | -     | 115    |
| y-Koordinate | 120       | -     | 35     |
| z-Koordinate | 130       | -     | 8      |
| Durchmesser  | 140       | -     | 10     |
| Tiefe        | 150       | -     | 8      |
| Loch         | 100       | False | True   |
| x-Koordinate | 110       | -     | 35     |
| y-Koordinate | 120       | -     | 115    |
| z-Koordinate | 130       | -     | 8      |
| Durchmesser  | 140       | -     | 10     |
| Tiefe        | 150       | -     | 8      |
| Gewinde      | 200       | False | False  |
| x-Koordinate | 210       | -     | -      |
| y-Koordinate | 220       | -     | -      |
| z-Koordinate | 230       | -     | -      |
| Durchmesser  | 240       | -     | -      |
| Tiefe        | 250       | -     | -      |
| Höhe         | 300       | 10    | 8      |
| Mulde        | 400       | False | False  |
| x-Koordinate | 410       | -     | -      |
| y-Koordinate | 420       | -     | -      |
| z-Koordinate | 430       | -     | -      |
| Tiefe        | 440       | -     | -      |
| Winkel       | 450       | -     | -      |
| Logo         | 500       | False | False  |
| x-Koordinate | 510       | -     | -      |
| y-Koordinate | 520       | -     | -      |
| z-Koordinate | 530       | -     | -      |
| Tiefe        | 540       | -     | -      |
| Winkel       | 550       | -     | -      |
| Geprüft      | 600       | False | False  |
| Loch         | 610       | False | False  |
| x-Koordinate | 611       | -     | -      |
| y-Koordinate | 612       | -     | -      |
| z-Koordinate | 613       | -     | -      |
| Durchmesser  | 614       | -     | -      |
| Tiefe        | 615       | -     | -      |
| Mulde        | 620       | False | False  |
| x-Koordinate | 621       | -     | -      |
| y-Koordinate | 622       | -     | -      |
| z-Koordinate | 623       | -     | -      |
| Tiefe        | 624       | -     | -      |
| Winkel       | 625       | -     | -      |
| Logo         | 630       | False | False  |
| x-Koordinate | 631       | -     | -      |
| y-Koordinate | 632       | -     | -      |
| z-Koordinate | 633       | -     | -      |
| Tiefe        | 634       | -     | -      |
| Winkel       | 635       | -     | -      |
| Lackiert     | 700       | False | False  |
| Farbe        | 710       | -     | -      |
| Breite       | 800       | 150   | 150    |
| Tiefe        | 900       | 150   | 150    |

```

<Portigungsauftrag package_Prefix:Portigungsauftrag xmlns:package="2.0" xmlns:
  <produkt ProduktID="Distanzscheibe1">
    <produktparameter Merkmal="material" Name="material" Datentyp="String">
      <produktparameter Merkmal="100" Name="hole" Datentyp="boolean">
        <input Wert="false"/>
        <output Wert="true"/>
      </produktparameter>
      <produktparameter Merkmal="110" Name="xCoordinate" Datentyp="int">
        <input Wert="-"/>
        <output Wert="35"/>
      </produktparameter>
      <produktparameter Merkmal="120" Name="yCoordinate" Datentyp="int">
        <input Wert="-"/>
        <output Wert="35"/>
      </produktparameter>
      <produktparameter Merkmal="130" Name="zCoordinate" Datentyp="int">
        <input Wert="-"/>
        <output Wert="8"/>
      </produktparameter>
      <produktparameter Merkmal="140" Name="diameter" Datentyp="int">
        <input Wert="-"/>
        <output Wert="10"/>
      </produktparameter>
      <produktparameter Merkmal="150" Name="depth" Datentyp="int">
        <input Wert="-"/>
        <output Wert="8"/>
      </produktparameter>
    </produktparameter>
    <produktparameter Merkmal="100" Name="hole" Datentyp="boolean">
      <input Wert="false"/>
      <output Wert="true"/>
    </produktparameter>
    <produktparameter Merkmal="110" Name="hole" Datentyp="boolean">
      <input Wert="false"/>
      <output Wert="true"/>
    </produktparameter>
    <produktparameter Merkmal="120" Name="hole" Datentyp="boolean">
      <input Wert="false"/>
      <output Wert="true"/>
    </produktparameter>
    <produktparameter Merkmal="200" Name="thread" Datentyp="boolean">
      <input Wert="false"/>
      <output Wert="false"/>
    </produktparameter>
    <produktparameter Merkmal="210" Name="xCoordinate" Datentyp="int">
      <input Wert="10"/>
      <output Wert="115"/>
    </produktparameter>
    <produktparameter Merkmal="220" Name="yCoordinate" Datentyp="int">
      <input Wert="10"/>
      <output Wert="35"/>
    </produktparameter>
    <produktparameter Merkmal="230" Name="zCoordinate" Datentyp="int">
      <input Wert="10"/>
      <output Wert="8"/>
    </produktparameter>
    <produktparameter Merkmal="240" Name="diameter" Datentyp="int">
      <input Wert="10"/>
      <output Wert="10"/>
    </produktparameter>
    <produktparameter Merkmal="250" Name="depth" Datentyp="int">
      <input Wert="10"/>
      <output Wert="8"/>
    </produktparameter>
    <produktparameter Merkmal="300" Name="height" Datentyp="int">
      <input Wert="10"/>
      <output Wert="8"/>
    </produktparameter>
    <produktparameter Merkmal="400" Name="trough" Datentyp="boolean">
      <input Wert="false"/>
      <output Wert="false"/>
    </produktparameter>
    <produktparameter Merkmal="410" Name="xCoordinate" Datentyp="int">
      <input Wert="420" Name="yCoordinate" Datentyp="int">
        <input Wert="430" Name="zCoordinate" Datentyp="int">
          <input Wert="440" Name="depth" Datentyp="int">
            <input Wert="450" Name="angle" Datentyp="int">
              <produktparameter Merkmal="500" Name="logo" Datentyp="boolean">
                <input Wert="false"/>
                <output Wert="false"/>
              </produktparameter>
              <produktparameter Merkmal="510" Name="xCoordinate" Datentyp="int">
                <input Wert="520" Name="yCoordinate" Datentyp="int">
                  <input Wert="530" Name="zCoordinate" Datentyp="int">
                    <input Wert="540" Name="depth" Datentyp="int">
                      <input Wert="550" Name="angle" Datentyp="int">
                        <produktparameter Merkmal="600" Name="checked" Datentyp="boolean">
                          <input Wert="false"/>
                          <output Wert="false"/>
                        </produktparameter>
                        <produktparameter Merkmal="610" Name="hole" Datentyp="boolean">
                          <input Wert="false"/>
                          <output Wert="false"/>
                        </produktparameter>
                        <produktparameter Merkmal="620" Name="trough" Datentyp="boolean">
                          <input Wert="false"/>
                          <output Wert="false"/>
                        </produktparameter>
                        <produktparameter Merkmal="630" Name="logo" Datentyp="boolean">
                          <input Wert="false"/>
                          <output Wert="false"/>
                        </produktparameter>
                        <produktparameter Merkmal="700" Name="painted" Datentyp="boolean">
                          <input Wert="false"/>
                          <output Wert="false"/>
                        </produktparameter>
                        <produktparameter Merkmal="710" Name="paint" Datentyp="string">
                          <input Wert="150"/>
                          <output Wert="150"/>
                        </produktparameter>
                        <produktparameter Merkmal="800" Name="width" Datentyp="int">
                          <input Wert="150"/>
                          <output Wert="150"/>
                        </produktparameter>
                        <produktparameter Merkmal="900" Name="depth" Datentyp="int">
                          <input Wert="150"/>
                          <output Wert="150"/>
                        </produktparameter>
                      </input>
                    </input>
                  </input>
                </input>
              </input>
            </input>
          </input>
        </input>
      </input>
    </input>
  </produkt>
</Portigungsauftrag package_Prefix:Portigungsauftrag>

```

Abbildung C.1: Beschreibung des Produktionsauftrags *Distanzscheibe 1* und zugehörige XML-Datei





