DESIGNING FUTURE FACTORIES

A novel approach to configuring production systems
by combining set-based and automated design

JOHANNES UNGLERT

DESIGNING FUTURE FACTORIES
A NOVEL APPROACH TO CONFIGURING PRODUCTION SYSTEMS BY COMBINING
SET-BASED AND AUTOMATED DESIGN

DISSERTATION

to obtain
the degree of doctor at the University of Twente,
on the authority of the rector magnificus
prof. dr. T.T.M. Palstra
on account of the decision of the graduation committee,
to be publicly defended
on Wednesday the 28th of February 2018 at 14:45

by

Johannes Markus Unglert
born on the 30th of April 1988
in Munich, Germany

This dissertation has been approved by the Supervisor:

Prof. dr. ir. F.J.A.M. van Houten

and the Co-supervisors:

Dr. ir. J. Jauregui-Becker

Dr. ir. S. Hoekstra

DESIGNING FUTURE FACTORIES

**A novel approach to configuring production systems by
combining set-based and automated design**

Johannes Unglert

Members of the graduation committee

Prof. dr. G.P.M.R. DeWulf (University of Twente, Chairman & secretary)

Prof. dr. ir. F.J.A.M. van Houten (University of Twente, Supervisor)

Dr. ir. J.M. Jauregui-Becker (University of Twente, Co-supervisor)

Dr. ir. S. Hoekstra (University of Twente, Co-supervisor)

Prof. dr. ir. J.I.M. Halman (University of Twente)

Prof. dr. ir. H. Zijm (University of Twente)

Prof. dr. ir. J. Post (University of Groningen)

Prof. dr. László Monostori (MTA SZTAKI Hungary)

This research was partly supported by the European Union in context of the project "Robust PlaNet: Shock-robust design of Plants their Supply Chain Networks" (7th Framework Programme, grant no. 609087).

Cover design © Johannes Unglert 2018

The illustration on the title page shows the connection of the three central concepts of this research: the production engineers, the software application to support the production engineers and the modular production system under design.

ISBN: 978-90-365-4487-0

DOI: 10.3990/ 1.9789036544870

Printed by Ipskamp Drukkers BV, Enschede, The Netherlands

No part of this work may be reproduced or transmitted for commercial purposes, in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted by the publisher.

SUMMARY

Production technology innovations introduced in the last decades have helped manufacturing companies to create the capabilities to produce new and complex products more efficiently. One type of system concepts that made this possible are Reconfigurable Manufacturing Systems (RMS). An important characteristic of RMS is that they enable the production system engineers to change the configuration of the production system. Hence, the capabilities and capacities of RMS can be adjusted in response to changing customer demand, which allows to re-use the system components, extend the lifecycle of these assets and maximize the system-wide value-added. Nevertheless, the design of system configurations bears a high complexity for the production engineers, since numerous configuration opportunities exist, each with its own implications. This so-called design space of RMS configurations can be heterogenic, large and difficult to grasp by the decision-makers. Every time the system is reconfigured, the system designers have to come up with suitable configuration designs, determine and evaluate their implications and eventually choose one of these solutions. In this challenging process, computational tools can relieve the system designers from tedious tasks by automating the design, analysis and evaluation of system configurations and thereby support the design process.

In this thesis I determine the basic objectives and required features of computational tools for automated design of reconfigurable production systems and propose a novel approach to automating their design. Subsequently, I evaluate the approach based on experimentation to determine its implications and conduct an empirical study to indicate its usefulness and applicability in industrial practice.

In addition to the number of configuration opportunities, the complexity and extent of the design task depends significantly on the possible number and types of the involved subsystems. This implies that a large number of design variables can exist, whose impact on the performance of the resulting systems cannot be stated generally and has to be determined for each specific case. Furthermore, it can be required to consider interdependencies between the variables of the design problem, as well as extrinsic pre-conditions for system configurations, such as space limitations associated to the existing factory infrastructure. An important implication with regard to these factors is that frame conditions for planning the future system - such as customer demand - are often subject to high uncertainty. Hence, the assumptions used as basis for developing the design are often difficult to forecast and justify. Individually and also collectively these circumstances can make the configuration design a complex endeavor.

Design automation software can be used to support the system designers in mastering these difficulties. By automating tedious tasks associated to the synthesis, analysis and evaluation of system configurations, the main benefits of such tools can be an increased effectiveness and efficiency of the configuration design process. As a result, decision-makers can get an understanding of possible alternative solutions and thus obtain well-performing systems. Moreover, the tools can reduce the time required for assessing a broad spectrum of

solutions. Yet, the development of design automation tools commonly requires significant efforts that have to be justified considering their prospective capabilities. In this context, the re-usability and versatility of the support tools play important roles.

This research discusses the link between the characteristics of RMS configuration design and design automation approaches to establish a framework of the relevant features and objectives that tools for automated design of RMS configurations should expose. The main objectives of such tools are efficiency and effectiveness of the automated generation, analysis and evaluation of system designs. The key features to realize these objectives are algorithms and user interfaces that make it possible to generate solutions for various formulations of the system design problem, flexibly integrating preference information for the solutions and integrally comparing the resulting solutions.

The innovative design automation tool developed here aims to realize the functionality and objectives of the framework by combining automated design, customizable visualization of the resulting design spaces and the rationale of set-based design, i.e. decision-making based on multiple alternatives. The system model and algorithms are tailored to the concrete problem case of a manufacturing company, which is presented in detail. The anticipated functionality of the developed software is explained subsequently. It enables the user to iteratively test and explore the implications of various ways of formulating the design problem and associated assumptions.

Following the presentation of the developed tool, its evaluation is described. The first part of the evaluation is focused on the tool's intrinsic characteristics with regard to the synthesized system designs. It examines the de-facto coverage of the potential design space by the algorithms, the implications of changed user input and the potential to suggest solutions for various problems. The results of this evaluation confirm the aspired effects of the approach, namely its capability to efficiently generate a broad spectrum of different solutions for different problem formulations that can be assessed and compared by the users. In addition, the evaluation results make it possible to identify improvement potentials with regard to the effectiveness of the design synthesis algorithms. The second part of the evaluation focuses on the extrinsic effects. An empirical study was designed to let industrial system designers determine the practical applicability and usefulness of the tool. For this purpose, the designers used the tool to solve two experimental cases, which represent realistic system design problems. After the experimental cases, qualitative data was collected by leading semi-structured interviews with the objective to document the system designers' opinion with regard to the applicability and usefulness of the tool. The designers stated that the tool would allow them to efficiently increase the number of configurations considered and to strategically approach the design of configurations. The results of these interviews indicate a high use value and applicability of the tool in industrial practice.

This research provides three major contributions. Firstly, the proposed reference framework makes it possible to evaluate existing and future support tools and detect their respective implications. Secondly, the support tool and its detailed description represent an innovative approach to support production engineers by means of design automation, which supports the users in determining well-performing system configurations and reducing the duration of the design process. Thirdly, the evaluation and validation of the tool with regard to the aspired improvements of efficiency and effectiveness of the design process allow a differentiated impression of the tool's strengths and opportunities for improvement. In this context, the innovative design of the empirical study for assessing the tool in industrial practice has to be highlighted, as such approach to evaluation has not been found in existing literature on support tools for production system design.

SAMENVATTING

Dankzij innovaties op het gebied van productietechnologie zijn maakbedrijven in staat nieuwe en complexe producten te produceren en de efficiëntie van productieprocessen te verhogen. Deze voordelen worden geboden door herconfigureerbare productiesystemen (RMS). Door het vermogen en de capaciteit van het systeem aan te passen aan de veranderende klantvraag, kunnen bij elk herconfiguratie nieuwe componenten in het systeem geïntegreerd worden. Daarbij kunnen ook oude systeemcomponenten herhaaldelijk en op verschillende manieren worden ingezet, waardoor de nuttige levensduur verlengd wordt. Desalniettemin brengt het configureren van herconfigureerbare productiesystemen een aanzienlijke complexiteit met zich mee, omdat er vaak een enorm aantal opties voor een configuratie bestaat (de zogenaamde *ontwerpruimte*). Bij elke herconfiguratie moeten systeemontwerpers relevante configuraties ontwikkelen, hun kenmerken evalueren en een geschikte oplossing vinden. In dit uitdagende proces kunnen softwaretools systeemontwerpers ondersteunen in het ontwikkelingsproces door langdurige en vervelende taken te vergemakkelijken met geautomatiseerde configuratiesynthese, analyse en beoordeling.

De hoofddoelen van dit proefschrift zijn drieledig: ten eerst worden de belangrijkste doelen en functies van design automation tools bepaald voor de ondersteuning van de RMS configuratietaak. Ten tweede wordt een nieuw hulpmiddel voor geautomatiseerde configuratiesynthese geïntroduceerd. Tot slot wordt de evaluatie van het nieuwe hulpmiddel uitgevoerd. Hierin worden de gevolgen voor de gebruiker als ook de praktische toepasbaarheid en de meerwaarde in detail besproken.

Naast het aantal configuratie-opties, kan ook de omvang van de configuratietaak aanzienlijk en uitdagend zijn. Deze omvang hangt voornamelijk af van het aantal en type subsystemen in de configuratie. Dit impliceert dat er een groot aantal ontwerpvariabelen zouden kunnen bestaan, waardoor de prestaties van de configuraties niet generiek kunnen worden voorspeld maar per configuratie moeten worden bepaald. Daarnaast kunnen er afhankelijkheden tussen ontwerpvariabelen en eisen voor configuraties bestaan, bijvoorbeeld als gevolg van bestaande fabrieksinfrastructuur. Bovendien bestaat er vaak veel onzekerheid over de randvoorwaarden voor het toekomstige systeem, zoals de klantvraag. Dit kan er toe leiden dat informatie die als basis voor de configuratie-ontwikkeling gebruikt wordt, moeilijk voorspeld kan worden, en daardoor moeilijker gerechtvaardigd kan worden. Individueel en in interactie kunnen deze voorwaarden de ontwikkelingstaak een complexe onderneming maken.

Computerapplicaties kunnen systeemontwikkelaars in zulke moeilijkheden ondersteunen. Door de automatisering van het systeemontwikkelingsproces kunnen systeemontwerpers makkelijk grip krijgen op alternatieve configuratie-opties en voor high-performance configuraties kiezen. Bovendien kunnen zulke computerapplicaties de duur van het proces voor het evalueren van alternatieven verkorten. Echter brengt de ontwikkeling van dergelijke hulpmiddelen vaak een aanzienlijke kostenpost met zich mee, die gerechtvaardigd

moet worden door de voordelen van deze tools. In deze context spelen de herbruikbaarheid en veelzijdigheid van hulpmiddelen een sleutelrol.

Dit onderzoek bespreekt de verbinding tussen de karakteristieken van het ontwerp taak van RMS configuraties en benaderingen om deze taak te ondersteunen. Op basis daarvan wordt een raamwerk voorgesteld voor de functies en doelen voor tool's die ondersteuning bieden voor het configureren van herconfigureerbare productiesystemen door geautomatiseerde design. De belangrijkste doelen zijn de efficiëntie en effectiviteit van de geautomatiseerde generatie, analyse en selectie van voorgestelde oplossingen. Sleutelfactoren om deze doelen te bereiken zijn tool-functionaliteiten waarmee oplossingen kunnen worden gevonden in verschillende formuleringen van het ontwerp probleem, evenals de flexibele integratie van voorkeuren voor specifieke oplossingen en het integraal vergelijken van oplossingen.

Op basis van dit raamwerk wordt een innovatief tool gepresenteerd, dat de beschreven functionaliteit en doelen te behalen streeft met een combinatie van geautomatiseerde en geïndividualiseerde oplossingsvoorstellen, evenals de visualisatie van de resulterende ontwerp mogelijkheden. Het concept weerspiegelt de logica van set-based ontwerp processen – oftewel, ontwerp processen die gebaseerd zijn op een aantal alternatieve oplossingen. Het ontwikkelde computer-tool, het systeemmodel en de algoritmen voor één specifieke toepassing worden in detail beschreven. Dit vormt de basis voor het beschrijven van de tool-functionaliteit, die het mogelijk maakt op iteratieve en verkennende wijze implicaties van verschillende eisen en doelspecificaties voor configuraties te testen.

Na de beschrijving van de tool vindt de evaluatie plaats, die is verdeeld in een intrinsiek en een extrinsiek onderdeel. In het intrinsieke deel wordt de tool onderzocht op verschillende kenmerken met betrekking tot het genereren van oplossingen, waaronder de feitelijke afdekking van de mogelijke ontwerp ruimte door de ontwikkelde algoritmen, het gedrag van de algoritmen met veranderde input van de gebruikers en het potentieel om oplossingen voor verschillende ontwerp problemen te vinden. De resultaten van deze evaluatie bevestigen de bedoelde effecten van de methode, namelijk het efficiënt genereren van een breed spectrum van verschillende oplossingen en deze toegankelijk maken voor evaluatie door de systeemontwikkelaars. Bovendien laten de resultaten ook toe om te identificeren waar kansen voor verbeteringen liggen in het proces voor het vinden van oplossingen. Het tweede deel van de evaluatie beschrijft een empirisch onderzoek met systeemontwerpers uit de industrie en is erop gericht de toepasbaarheid en nut van de tool in de praktijk te bepalen. Hiertoe zijn verschillende testcases ontworpen, waaraan de ontwerpers hebben gewerkt met het prototype van de tool. Na het werk aan de cases is de ervaring met de tool vastgelegd door de systeemontwikkelaars te interviewen. Volgens de ontwikkelaars maakt de tool mogelijk om efficiënt een groot aantal oplossingen te evalueren en de systeemconfiguratie strategisch te benaderen. De kwalitatieve resultaten tonen aan dat de methode in de bedrijfspraktijk nuttig kan zijn en waarschijnlijke goede toepasbaar is.

De belangrijkste bijdrage van dit onderzoek is driedig. Ten eerste maakt het raamwerk met de referentiecriteriën voor computerapplicaties het mogelijk bestaande en nieuwe ondersteuningsinstrumenten te vergelijken en om hun gevolgen te ontdekken. Ten tweede biedt de gedetailleerde beschrijving van de ontwikkelde ondersteuningsmethode een goede uitleg van het werkingsprincipe. Ten derde biedt de evaluatie van de tool een gedifferentieerde indruk van het potentieel van het hulpmiddel, evenals mogelijke benaderingen om het te verbeteren. In dit verband kan ook de innovatieve benadering van de praktische validatie van de gewenste effecten genoemd worden, aangezien een dergelijke test met bestaande instrumenten in de gewenste toepassingsomgeving nog niet is gedocumenteerd.

ZUSAMMENFASSUNG

Dank Produktionstechnologie-Innovationen sind Unternehmen zunehmend in der Lage neue und komplexe Produkte zu produzieren, sowie die Produktionseffizienz zu erhöhen. Rekonfigurierbare Produktionssysteme (RPS) sind ein Typus neuartiger Fertigungssysteme, die es den Gestaltern ermöglichen die Systemkonfigurationen zu wechseln. Durch Anpassung der Systemkapazitäten und –Ressourcen an eine sich verändernde Kundennachfrage können Systemkomponenten wiederholt und auf verschiedene Arten eingesetzt werden. Ferner bieten diese Systeme die Möglichkeit, neue Fertigungsressourcen zu integrieren, die Nutzungsdauer der Ressourcen zu erhöhen und ihre Wertschöpfung zu maximieren. Nichtsdestotrotz bringt der Gestaltungsprozess von Konfigurationen eine erhebliche Komplexität mit sich, da oftmals zahlreiche und heterogene Konfigurations-Möglichkeiten bestehen (der sog. „design space“). Bei jeder Rekonfiguration müssen die Entscheider relevante Konfigurationen entwickeln, deren Charakteristika bewerten und eine passende Lösung wählen. In diesem herausfordernden Prozess können Software Tools dabei helfen, die Systemdesigner von monotonen und fehlerträchtigen Aufgaben zu befreien, indem sie die Konfigurationssynthese, Leistungsbestimmung und –Bewertung automatisieren und dadurch den Entwicklungsprozess unterstützen.

Diese Dissertation zielt darauf ab, wesentliche Ziele und dafür benötigte Funktionalitäten von Computeranwendungen zur automatisierten Generierung von Konfigurationen zu ermitteln. Darauf basierend wird ein neuer Ansatz zur automatisierten Lösungssynthese vorgestellt. Schließlich wird eine Evaluierung der Methodik vorgenommen, um die impliziten Effekte sowie die Anwendbarkeit und den erwarteten Nutzen in der industriellen Praxis genauer beschreiben zu können.

Beim Entwickeln von RPS-Konfigurationen kann neben der Anzahl der Konfigurationsmöglichkeiten auch der Umfang der Entwicklungsaufgabe eine Herausforderung darstellen, die noch größer wird, wenn mehrere Subsysteme zugleich konfiguriert werden sollen. In diesem Fall kann eine erhebliche Anzahl an Konfigurations-Variablen bestehen, deren Einfluss auf die Leistung der resultierenden Systeme nicht allgemein vorhergesagt werden kann und spezifisch ermittelt werden muss. Daneben können Abhängigkeiten zwischen Konfigurations-Variablen vorliegen, die ebenso im Entwicklungsprozess berücksichtigt werden müssen wie Anforderungen für Konfigurationen, die beispielsweise durch die bestehende Fabrikinfrastruktur bedingt sind. Zudem besteht oft eine große Unsicherheit in Bezug auf die Rahmenbedingungen für das zukünftige System. Dies kann dazu führen, dass Informationen nur schwer vorhersagbar sind – wie zum Beispiel die Kundennachfrage – und damit nicht ausreichend gerechtfertigt werden können, obwohl sie als Grundlage für die Entwicklung der Konfigurationen benötigt werden. Im Einzelnen und im Zusammenspiel können diese Rahmenbedingungen die Entwicklungsaufgabe maßgeblich verkomplizieren.

Computeranwendungen können die Entscheider in dieser Situation unterstützen, indem sie Aufgaben im Systementwicklungsprozess automatisieren und sich dadurch positiv auf

die Prozesseffektivität und -effizienz auswirken. Dadurch können Systementwickler ein Verständnis alternativer Konfigurationsmöglichkeiten erlangen und performante Konfigurationen wählen. Zudem können solche Anwendungen die Dauer des Prozesses zur Bewertung von Alternativen reduzieren. Um solche Hilfsmittel zu entwickeln ist jedoch oftmals ein beträchtlicher Aufwand nötig, der dem erwarteten wirtschaftlichen Nutzen des Tools gegenübergestellt werden und gerechtfertigt sein muss. In diesem Zusammenhang spielt die Wiederverwendbarkeit und Vielseitigkeit der Tools eine zentrale Rolle.

Diese Forschungsarbeit diskutiert die Schnittstellen der Entwicklung von Konfigurationen des Produktionssystems und Computeranwendungen, die dabei unterstützen sollen. Ferner wird ein Rahmenkonzept für die Funktionen und Ziele vorgestellt, die Support Tools zur automatisierten Konfiguration von RPS aufweisen sollten. Hauptziele sind dabei die Effizienz und Effektivität der automatischen Generierung, Analyse und Bewertung von Lösungsvorschlägen. Wesentliche Faktoren zur Realisierung dieser Ziele sind Toolfunktionalitäten, die Lösungen zu verschiedenartigen Formulierungen des Konfigurationsproblems auf effiziente Weise vorschlagen können. Dafür müssen Entscheider in der Lage sein, ihre Anforderungen und Präferenzen flexibel in das Tool integrieren und vorgeschlagene Lösungen integral vergleichen zu können. Hierfür sind darauf abgestimmte Algorithmen und Funktionalitäten erforderlich.

Basierend auf diesem Rahmenkonzept wird ein innovativer Ansatz vorgestellt, der darauf abzielt, die beschriebene Funktionalität und Ziele durch eine Kombination von automatisierten und individualisierbaren Lösungsvorschlägen sowie der Visualisierung der resultierenden Gestaltungsmöglichkeiten zu erreichen. Dabei spiegelt das Konzept die Logik eines set-basierten – d.h. auf Basis von mehreren alternativen Lösungen gestalteten – Entscheidungsprozesses wieder. Das entwickelte Computerprogramm, das Modell des Produktionssystems und die Algorithmen für einen konkreten Anwendungsfall werden detailliert beschrieben. Dies stellt die Basis für die Erläuterung der Tool-Funktionalität dar, die dadurch charakterisiert ist, dass sie ein iteratives und exploratives Testen verschiedener Anforderungen und Zielspezifikationen für Konfigurationen ermöglicht.

Im Anschluss daran wird die Evaluation beschrieben, welche in einen intrinsischen Teil und einen extrinsischen Teil gegliedert ist. Im intrinsischen Teil wird die Methodik auf mehrere Charakteristika bezüglich der Lösungsgenerierung untersucht, genauer die de-facto Abdeckung des möglichen Lösungsraums durch die Algorithmen, das Verhalten des Tools bei geänderten Nutzereingaben sowie das Potenzial des Tools, Lösungen für verschiedenartige Probleme zu finden. Die Ergebnisse dieser Evaluierung bestätigen die angestrebten Effekte der Methode, nämlich auf effiziente Weise ein breites Spektrum unterschiedlicher Lösungen zu generieren und einer Bewertung zugänglich zu machen. Zudem erlauben die Resultate auch die Identifikation möglicher Verbesserungspotenziale in Bezug auf den Prozess zur Lösungsfindung. Der zweite Teil der Evaluation beschreibt eine empirische Studie mit Systemdesignern aus der Industrie, die zum Ziel hat, die praktische Anwendbarkeit und den Nutzen des Tools zu bestimmen. Dazu wurden verschiedene Testfälle gestaltet, die die Designer mit dem Tool-Prototyp bearbeiten sollten und Fragebögen konzipiert, die die Erfüllung der beabsichtigten Effekte des Tools aus der Perspektive der Entscheider nach der Lösung der Fälle erfassen. Nach Einschätzung der Systemdesigner ermöglicht das Support Tool, auf effiziente Art eine hohe Anzahl an Lösungen zu evaluieren und einen strategischen Ansatz zur Systemkonfiguration zu verfolgen. Die Resultate deuten schließlich auf einen hohen Nutzwert und die Anwendbarkeit der Methodik in der unternehmerischen Praxis.

Im Ergebnis liefert diese Forschungsarbeit aus wissenschaftlicher Sicht drei wesentliche Beiträge: Erstens ermöglicht das vorgestellte Konzept der Referenzkriterien für computer-gestützte Support Tools, bestehende und zukünftige Anwendungen zu vergleichen und deren Vor- bzw. Nachteile festzustellen. Zweitens erlaubt die detaillierte Beschreibung des entwickelten Support Tool ein tiefes Verständnis von dessen Funktionsprinzip. Drittens gewährt die Evaluierung der Methodik einen differenzierter Einblick in die Potenziale und mögliche Ansätze für weitere Verbesserung der vorgestellten Anwendung. In diesem Zusammenhang soll auch der innovative Ansatz zur praktischen Validierung der angestrebten Effekte Erwähnung finden, da in Bezug auf bestehende Tools eine solche Evaluierungsmethodik im angestrebten Anwendungsumfeld bisher nicht dokumentiert wurde.

ACKNOWLEDGMENTS

This is it! The booklet lying in front of you is the result of roughly four years of work at the University of Twente. I would like to express my sincere gratitude to a couple of people that supported me throughout these years and without whom this work would have been hardly possible.

First and foremost, I want to thank Fred van Houten for giving me the chance to dive into this exciting research field and for giving me the freedom to bring in my own ideas and develop myself professionally. Your openness to discuss new ideas and provide support in difficult moments gave me the confidence to fully make this research my own and thus cannot be appreciated enough. Not less, I want to thank my two co-supervisors, Juan Jauregui-Becker and Sipke Hoekstra, for the commitment they have shown throughout the past years. The combination of your enthusiastic and proactive, yet also patient and indulgent company played a key role in allowing me to master the trajectory in a motivated way. Thanks are also owed to Hans Tragter. Without your previous work on the tool, the introduction to programming and collaboration on the software tool I probably would not have managed to obtain a result of this standard.

Further thanks go out to the instructors and fellow students of the Summer School of Engineering Design Research and the International Spring School on Systems Engineering, which I had the pleasure to attend. The opportunities to present one's own research in-depth and receiving feedback from such reputable researchers and peers provided an invaluable contribution to advancing my research and the social activities surrounding the workshops provided to a quite enjoyable atmosphere.

Further credits go to all colleagues with whom I enjoyed collaborating across European borders in the Robust-PlaNet research project and in particular to Dávid Gyulai, Massimo Manzini, Mario Smink, Martin Karelse and Jeroen ter Mate for their friendly and productive cooperation. The project made me experience the sometimes slightly intangible benefits of the European project and fostered my belief that an integrated and united Europe provides the best circumstances for its citizens to innovate and strive together. In this respect, I also want to express my gratitude for the funding of my research by the European Union.

Additionally, I would like to express my sincere gratitude to all colleagues at the department DPM that made working at the UT a pleasant experience. Roberto and Alberto, special thanks for your unselfish support and friendship! Vangelis, thanks for your friendship and offering shelter to a German in Greece and during my trips to the Netherlands! Thanks for the fun memories and football competitions also goes to the all members of the infamous *OPM Ultras*, namely captains Monique, Pieter, Moreno, Patrick, Robert-Jan, Willem and Jan-Jaap. Thanks in particular to all people who accompanied and supported me throughout the past years: Inge d.S., Jörg, Inge H., Anne-Marie, Wieteke, Wienik, Julia, Maarten B., Eric, Tox, Maarten E., Hans, Fredje, Steven, Rick, Jorge, Adriaan, Jos T., Jos d.L., Ellen, Rick, Katja and Merel.

Neben den Leuten, die ich in den Niederlanden und andernorts im Zusammenhang mit meiner Arbeit kennen lernen durfte, bin ich froh darüber einigen Menschen in meinem Leben bereits davor begegnet zu sein. Gute Freunde, die einem helfen seinen Blick immer wieder auf das Wesentliche und Schöne im Leben zu lenken, sind unbezahlbar! Jan, Philipp, Verena, Saskia, Andi, Sebastian, Philipp, Felix, Vroni, Matze, Ines und alle anderen Landsberger: vielen Dank dafür dass wir uns noch so häufig sehen und es schaffen, zusammen viele erinnerungswürdige Momente zu erleben. Ebenso möchte ich den Münchner Stapelfahrer-Kollegen Kai, Max, Chris und Sancia danken. Obwohl – oder gerade weil – wir ortsbedingt zuletzt nicht mehr so häufig zusammenfinden konnten, fand ich es jedes Mal etwas Besonderes. Vielen Dank dafür! Mindestens genauso freue ich mich darüber, dass es auch die ehrenwerten Mitglieder der Runde der Reutlinger Metaxa-Connaissseure – Fabi, Simon, Jörg – immer wieder schaffen zusammen zu kommen und das hoffentlich auch in Zukunft fortsetzen werden.

Mein größter Dank gilt jedoch meiner Familie. Mama, Papa, vielen Dank für eure niemals endende Unterstützung. Euer Beitrag zu meiner Freude am Leben ist nicht in Worte zu fassen! Susi und Kathi, ich bin sehr froh darüber euch zu haben.

CONTENTS

I	RESEARCH CLARIFICATION	1
1	TOWARDS EXPLOITING THE FLEXIBILITY OF PRODUCTION SYSTEMS	3
1.1	Manufacturing efficiency: past and present	4
1.2	Changeable and reconfigurable production systems	6
1.3	Issues of designing production systems	7
1.4	Computational support approaches	7
1.5	Research motivation	8
1.6	Computational design synthesis	9
1.7	Research plan	11
1.7.1	Research objectives	11
1.7.2	Research approach and questions	11
1.7.3	Research context: The Robust PlaNet project	13
1.8	Thesis outline	14
II	DESCRIPTIVE STUDY I	15
2	FRAME OF REFERENCE	17
2.1	Production systems	17
2.1.1	Design characteristics	17
2.1.2	General engineering design approaches	24
2.1.3	Dedicated theories of production system design	26
2.1.4	Managerial issues of production system design	27
2.2	Reconfigurable manufacturing systems	29
2.2.1	Design characteristics	29
2.2.2	Design method	31
2.2.3	Managerial issues of RMS design	32
2.3	Computational design support	35
2.3.1	Types and characteristics of support tools	36
2.3.2	Design space exploration	37
2.3.3	Managerial issues of design automation tools	38
3	AUTOMATING RMS DESIGN: FUNCTIONAL REQUIREMENTS AND APPROACHES	41
3.1	Functional requirements for design automation tools	41
3.1.1	Suitable RMS design phases for design automation	41
3.1.2	RMS-related requirements for design automation tools	42
3.1.3	Functional framework for tools automating RMS design	44
3.2	Support tools for automated production system design	46
3.3	Concluding remarks	52
III	PRESCRIPTIVE STUDY	53
4	THE DSET – A TOOL FOR AUTOMATED, SET-BASED DESIGN OF RMS	55

4.1	Problem definition and scope of application	55
4.1.1	Problem definition	55
4.1.2	Scope of application	58
4.2	Implementation and features	58
4.2.1	Design problem model	60
4.2.2	Algorithmic design procedures	60
4.2.3	Interfaces and software specifics	68
4.2.4	System configuration properties and requirements	73
4.2.5	Usage of the DSET	74
4.3	Envisaged fulfillment of the support tool objectives	76
4.4	Discussion	77
4.5	Conclusions	78
IV	DESCRIPTIVE STUDY II	79
5	EVALUATION OF THE DSET'S CHARACTERISTICS	81
5.1	Research approach	81
5.2	Coverage of theoretical solutions by the synthesis procedures	82
5.2.1	Variety of solution characteristics	82
5.2.2	Variety of solutions	85
5.2.3	Discussion	90
5.3	Impact of changed user specifications	91
5.3.1	Varying problem information	91
5.3.2	Varying design requirements	93
5.3.3	Varying performance requirements	94
5.3.4	Discussion	95
5.4	Comparing solution spaces of various problems	96
5.4.1	System configurations with strategic capacity reserve	97
5.4.2	System configurations with low cost	102
5.4.3	Discussion	108
5.5	Conclusion	109
6	EVALUATION OF THE DSET IN PRACTICE	111
6.1	Research approach	111
6.1.1	Preparation of the experimental session	111
6.1.2	Experimental session (data collection I)	113
6.1.3	Data analysis I and preparation of the verification session	113
6.1.4	Verification and validation session (data collection II)	114
6.1.5	Data analysis II	114
6.2	Results	115
6.2.1	Results of evaluating the key features	115
6.2.2	Results of evaluating the complementary effects	115
6.2.3	Results of evaluating the session realism	116
6.3	Discussion	116
6.3.1	Key features of the application	116
6.3.2	Complementary effects of the features	118
6.3.3	Data availability and session realism	120
6.4	Conclusions	121
7	VALORIZATION OF THE DSET	123
7.1	Estimating the DSET's effects	123

7.2	Embedding the DSET in a workflow	125
7.2.1	Assembly cell configuration tool	125
7.2.2	Production planning and simulation tool	125
7.2.3	Reconfiguration planning tool	126
7.2.4	Complementary tool workflow	126
7.3	Extending the DSET's capabilities	127
7.4	Conclusions	129
8	SUMMARY, DISCUSSION AND CONCLUSION	131
8.1	Findings of the research	131
8.1.1	Accomplishment of the support tool objectives	131
8.1.2	Answers to the research questions	133
8.1.3	Accomplishment of the research objectives	134
8.2	Contributions and implications	135
8.3	Limitations	137
8.4	Future research	138
8.5	Conclusions	139
8.6	Research reflection	140
	BIBLIOGRAPHY	143
V	APPENDIX	153
A	APPENDIX	155
A.1	Scenario after realizing the visions of Industry 4.0	155
A.2	Excursus: Industry 4.0 state of the art	156
A.3	Production system changeability and related terminology	157
A.4	Example of product portfolio and suitable system types	159
A.5	Excursus: Industrial practice of production system design	160
A.6	Production system model details	162
A.7	Pareto-efficiency feature	165
A.8	Properties and analysis of the synthesized system configurations	166
A.9	Experimental data	173
A.10	Example for configuration opportunities	175
A.11	Evaluation introduction	176
A.12	Scenarios	179
A.13	Questions asked during the evaluations	180
A.14	Evaluation case solutions	182

LIST OF FIGURES

Figure 1	Constituent elements of production systems (based on [3])	3
Figure 2	Example for production system reconfigurations (left) and design spaces of reconfigurable systems (right)	6
Figure 3	Extent of automation inherent in optimization and simulation (based on [30]).	8
Figure 4	Extent of automation inherent in computational design synthesis . .	10
Figure 5	Point-based design, concurrent design and set-based concurrent design (based on based on [40, 41])	10
Figure 6	Elements in scope and rationale of the thesis	11
Figure 7	Main phases of the design research methodology (based on [45]) . .	12
Figure 8	Research types of the design research methodology (based on [45]) .	13
Figure 9	Example for delimiting production systems in various ways	18
Figure 10	Change-over related capacity losses	19
Figure 11	Different logistical implications of the production process types . . .	20
Figure 12	Example of an assembly tree (based on [49])	20
Figure 13	Organization principles of manufacturing systems	22
Figure 14	Example for capacity requirement and resource selection in the context of material flows	23
Figure 15	Left: Engineering design methodology (based on [52, 55]); right: Generic design activities (based on [30])	24
Figure 16	Intersecting product and production system development waterfalls (based on [58])	25
Figure 17	Left: Decomposed production system design (based on [23]); Right: Holistic production system design (based on [4])	27
Figure 18	Interdependencies between facility design and production system configurations (based on [18, 69, 70])	28
Figure 19	Characteristics of system configuration design problems (based on [76, 77])	30
Figure 20	Possible and produced spaces of routine (left), innovative (center) and creative design (left) (based on [79])	31
Figure 21	Generic RMS design method (based on [74])	31
Figure 22	Activities and phases in the lifecycle of reconfigurable production systems (based on [4, 74, 76])	32
Figure 23	Example for partitioning three elements (left); Comparison of Stirling combinations with exponential growth.	33
Figure 24	Flexible, reconfigurable and dedicated manufacturing systems; Left: Capacity and system cost (based on [18]); Right: Production volume and product variety (based on [16])	34
Figure 25	Effect of reconfiguration on productivity (based on [18])	35

Figure 26	Design preference expressed in terms of requirements (left) and objectives (right)	36
Figure 27	Visual exploration of the design space (based on [95])	37
Figure 28	Development activities for design automation tools (based on [97]) .	38
Figure 29	Profit/loss during lifecycles of RMS and support tools (based on [99])	40
Figure 30	Functional framework for design automation tools for the S&A steps of RMS design	44
Figure 31	Reconfigurable cell architecture	56
Figure 32	System configuration problem	57
Figure 33	Input and output of the DSET	58
Figure 34	Activities and milestones of the DSET-based design process	59
Figure 35	Example for input and results of the three design-rule sets	61
Figure 36	Solutions generated with default scaling factors (left) and constrained factors (right)	68
Figure 37	Interface for initializing synthesis runs and specifying requirements	70
Figure 38	Customizable interface for exploring, comparing and selecting system configurations	71
Figure 39	Customizable graphs for evaluating the generated solutions	72
Figure 40	Intended process flow of the DSET	75
Figure 41	Rationale of the DSET	75
Figure 42	Rationale of the DSET	76
Figure 43	Synthesis probabilities of individual solutions	78
Figure 44	Observation frequency of the distinct system configurations of the product-family rule set in the TCS (x-axis) and CCS (bars) in E2 (top graph) and E3 (bottom graph)	84
Figure 45	Maximum (left) and minimum property values (right) observed in the synthesis runs (product-family rule set)	86
Figure 46	Maximum (left) and minimum property values (right) observed in the synthesis runs (uni-cell rule set)	87
Figure 47	Observation frequency of the distinct system configurations of the uni-cell and multi-cell rule sets in the TCS (x-axis) and CCS (bars) in E4 (top graph) and E5 (bottom graph)	88
Figure 48	Maximum and minimum property values observed in the synthesis runs (multi-cell rule set)	89
Figure 49	Investment and utilization of the Pareto-selected solutions (5P-PI1)	99
Figure 50	Investment and utilization of the manually selected solutions (5P-PI1)	99
Figure 51	Investment and utilization of the Pareto-selected solutions (5P-PI2)	101
Figure 52	Investment and utilization of the manually selected solutions (5P-PI2)	101
Figure 53	Number of cells and total cost of the Pareto-selected solutions (25P-PI1)	103
Figure 54	Number of cells and total cost of the manually selected solutions (25P-PI1)	103
Figure 55	Investment and logistics cost of the Pareto-selected solutions (25P-PI1)	104
Figure 56	Investment and logistics cost of the manually selected solutions (25P-PI1)	104
Figure 57	Investment and logistics cost of the Pareto-selected solutions (25P-PI2)	106
Figure 58	Investment and utilization of the manually selected solutions (25P-PI2)	106
Figure 59	Investment and logistics cost of the Pareto-selected solutions (25P-PI2)	107

Figure 60	Investment and logistics cost of the manually selected solutions (25P-PI2)	107
Figure 61	Phases and main activities of the practical evaluation	112
Figure 62	Input information for the cases of the empirical evaluation	112
Figure 63	Comparison of the established design process and the DSET	124
Figure 64	Workflow integrating the decision-support tools	127
Figure 65	Effectiveness-related aspects at a glance	132
Figure 66	Allocation of academic contribution (based on [127])	135
Figure 67	Preference information coupling points (based on [30, 97])	137
Figure 68	Comparison of the number publications on Industrie 4.0/Industry 4.0	156
Figure 69	Changeability classification of production systems (based on [70])	157
Figure 70	Example for Pareto quality feature	165
Figure 71	Example for products (left) and production processes (right)	176
Figure 72	Reconfigurable production system concept	177
Figure 73	Three possible configuration types	178

LIST OF TABLES

Table 1	Overview of the structure, chapters and contents of the thesis	14
Table 2	Problem aspects and support features of DSS for production system design	48
Table 3	Steps and design rules of the product-family configuration type . . .	64
Table 4	Steps and design rules of the uni-cell configuration type	65
Table 5	Steps and design rules of the multi-cell configuration type	66
Table 6	Properties of synthesized production system configurations	74
Table 7	Profile of experiment 1	83
Table 8	Profile of experiment 2	85
Table 9	Profile of experiment 3	85
Table 10	Profile of experiment 4	86
Table 11	Profile of experiment 5	89
Table 12	Results of experiments 1 to 5	90
Table 13	Profile of experiment 6	91
Table 14	Original and varied input data of experiment 6	92
Table 15	Resulting system configurations in the original (left) and varied case (right)	92
Table 16	Profile of experiment 7	93
Table 17	Resulting system configurations of the original (PS_1 to PS_7) and the varied case (PS_1)	93
Table 18	Profile of experiment 8	94
Table 19	Resulting system configurations of the original (PS_910 to PS_1414) and varied case (PS_1414)	94
Table 20	Evaluation experiments for comparing solution spaces	96
Table 21	Scaling factors for the product-family rule set (columns 1-6) and the uni-cell and multi-cell rule set (rightmost column)	97
Table 22	Profile of experiment 9 (5P-PI1)	98
Table 23	Profile of experiment 10 (5P-PI2)	100
Table 24	Profile of experiment 11 (25P-PI1)	102
Table 25	Profile of experiment 12 (25P-PI2)	105

ACRONYMS

CDS	Computational design synthesis
CCS	Computationally-synthesized configuration space
CMS	Cyber-manufacturing system
FAG	Functional assembly group
DRM	Design research methodology
DSE	Design space exploration
DSET	Design space exploration tool
DSS	Decision-support systems
GUI	Graphical user interface
RMS	Reconfigurable manufacturing system
SBCE	Set-based concurrent engineering
S&A	Selection and allocation
SE	Systems engineering
SoS	System of systems
TCS	Theoretical configuration space

Part I

RESEARCH CLARIFICATION

*Automation does not need to be our enemy.
I think machines can make life easier for men,
if men do not let the machines dominate them.*

— John F. Kennedy [1]

Mutatis mutandis, this statement represents a reply by the former US president John F. Kennedy to a question asked by a worried person over fifty years ago, at a time when the topic of automation started to become important in manufacturing. The debate on the implications of automation has continued to this day and makes the quote appear more recent than ever, although in a slightly different context. While automation traditionally has been used to support people in carrying out exhausting physical processes, the rise of computer technology has made it possible to use automation also to support people in performing complex cognitive tasks. In the context of manufacturing, these two forms of support can be used in the physical production of goods and determining the design of the production systems. Both aspects – but particularly the latter – will be principal topics in the following chapters.

TOWARDS EXPLOITING THE FLEXIBILITY OF PRODUCTION SYSTEMS

Manufacturing represents one of the most important commercial activities in the European Union (EU28). As of 2014, nearly 30 million people worked in manufacturing companies with a total turnover of slightly more than 7 trillion Euros [2]. This thesis focuses on companies producing discrete and durable products such as machinery and equipment, vehicles, transport equipment and fabricated metal products. These products generated almost one third of these turnovers (31,3%), making them vital factors for the wealth and development of many EU-countries.

Manufacturing systems – synonymous – production systems combine humans, machinery and equipment to create the products mentioned above. They are connected by flows of material and information, such as visualized in Figure 1 [3]. Manufacturing companies can therefore be described as production systems at various levels of aggregation, e.g. considering an entire factory, a department within a factory or even an individual work-station. Each system is characterized by the raw materials, energy and information about the products and their demand, which are used as input [3]. The output of the systems is material, mainly finished goods and scrap, and information such as measures of system performance. Independent of the level of consideration, the design of the production systems significantly affects the cost of the resulting products, as well as the quality and speed at which products can be produced. In this context, production technology is key to efficient processes that allow to produce products that reach the customers on time. Given the impact of production systems on the price, quality and availability of consumable and durable products and the resulting societal relevance of manufacturing companies, the design of production systems has to be carefully planned.

Figure 1: Constituent elements of production systems (based on [3])

Two terms often encountered in the literature dealing with products and production systems are design and development. The difference between the two concepts is defined as follows [4]:

- Design involves all tasks, starting with the initial definition of the problem, identification of the objectives and proposal of alternative solutions. After evaluating the suggested solutions, one of the candidate solutions should be selected and further detailed to obtain the result, which is the description of the system to be.
- Development includes the design and additionally the implementation, that is the building and industrialization, thereby spanning across a larger part of the system lifecycle than just design.

Therefore, design refers rather to the problem-solving process, whereas development additionally encompasses the realization of the determined solutions. To explain the objectives of production system design, Section 1.1 details the motivation for new approaches to production by outlining the main historical developments in the manufacturing industry, including the current challenges. Section 1.2 introduces changeability as an important characteristic of manufacturing systems and describes the implicit opportunities for production system designers. Because the design of production systems can prove difficult, typical issues are explained in Section 1.3. As a means to address these issues, the main computational support principles in design are presented in Section 1.4. Based on this foundation, Section 1.5 describes the motivation for performing research into the relevant software tools to support engineers in designing the factories of the future. Section 1.6 describes the proposed support approach and motivation of the research, followed by the research plans in Section 1.7 and the outline of the thesis in Section 1.8.

1.1 MANUFACTURING EFFICIENCY: PAST AND PRESENT

Since the beginning of industrialization, efficiency gains were an important motivation to adopt new approaches to production. Innovation in technology and management acted as enabler and driver for new production system paradigms. For taking new approaches to the design of production systems and thereby contributing to the unrivaled standard of material wealth we find ourselves in nowadays, the designers of the production systems deserve recognition.

The first significant increase of the efficiency of production processes marked the beginning of the first industrial revolution in the 18th century. Mechanical production processes to make products such as textiles or flour could be facilitated by using the power of steam and water [5]. As a result, the power formerly provided by workers or draft animals could be generated more cheaply, thereby leading to a greater productivity of the supported processes, which in turn enabled the producers to offer products at lower prices.

The second industrial revolution started around the 1870s and enabled mass production of commodities by means of novel principles for organizing work. Applied first in Chicago's cattle slaughterhouses and later also in the production of Henry Ford's famous Model T, the production steps were systematically separated and re-organized along conveyor belts. This transformation symbolized the birth of the production line and made it possible to create specialized workstations for specific production tasks along the line that could execute the respective production steps more efficiently. Additionally, the automated transport from one workstation to the next made manual transports redundant and further reduced the required efforts.

Around 1970, novel approaches to control manufacturing machines were increasingly adopted and led to the third industrial revolution. Instead of using workers to control the machines that executed the production processes, electronic control units – also described

as Programmable Logic Controllers – made it possible to automate the execution of simple tasks, resulting in further gains of process efficiency. Over time, the described organization principles and concepts of automated control became more mature and were adopted particularly in high-wage countries, where the automation of tasks performed by human workers promised – and still promises – significant savings while improving production capabilities [6]. An important factor in this regard is that the investments required for automating manufacturing processes are strongly determined by the prices of the automation components, embodied by the control units, the sensors and the actuators. The sensors enable the control unit to determine the current state of the system and the next steps to be performed by the actuators. As a result, control units can automatically take care of the execution of operation sequences. In this context, a remarkable development was that the ratio of computational performance to cost has exploded since 1970 [7]: until 2003 the number of instructions a computer system can perform per dollar invested grew by a factor of 164 million; the number of Megabits that can be stored per cubic centimeter increased by a factor of 77.500. Even though no scientific studies could be found with more recent numbers for these indicators, the trends of miniaturization and higher performance-cost ratios have likely prevailed until now. Furthermore, the sensor prices have dropped considerably, with a reduction of average prices from 0,66 to 0,40 US dollars per sensor alone between 2010 and 2015 [8]. At the same time novel control architectures for production systems such as those presented in [9] help to simplify the set-up of robots as actuators, reducing the efforts needed for automating tasks and thereby increase the economic viability of automated process execution. In combination, these factors have made the automation of production processes more affordable, allowed for faster and smaller controls and eventually enabled cheaper production throughout the last decades.

As of today, the next major steps in manufacturing technology and associated efficiency gains are expected to be realized by integrating production systems and the internet, making possible new system concepts described as Cyber-physical Manufacturing Systems (abbr. CMS) or – synonymous – cyber-physical production systems [10]. Initiatives promoting the research and development with regard to CMS can be found particularly in highly-industrialized countries such as the United States (Industrial internet of things), the Netherlands (Smart Industry) and Germany (Industrie 4.0). Even though slight differences exist concerning the formulation of the initiatives' objectives, they strive for the same results [11]: integrating information and communication technology with manufacturing and automation, eventually enabling manufacturing companies to produce customized products profitably even as one-of-a-kind solutions [5]. While this motive represents the logical extension of the effects of the previous revolutions, CMS promise benefits for companies and their customers beyond cheaper products, as illustrated in Appendix A.1. The impact of CMS is designated between 900 billion and 2,5 trillion US dollars in worldwide cost savings for manufacturing companies until the year 2025 [12, 13]. In CMS, the networked machines and other resources in the factory allow for high-speed communication, coordination and control of the system, yet, the actual flexibility of the system is limited by the specific physical and organizational structure of the shop-floor. Therefore, the information network is relevant for operating and managing the systems, whereas the systems have to be designed in ways that allow for their flexible usage in the first place. Despite the growing research interest, the state-of-the-art and the current level of penetration of CMS in industrial practice remains unclear (see Appendix A.2 for a brief discussion). Nevertheless, many manufacturing companies seem to be aware of the potential benefits they can yield with cyber-physical production systems. The results of a survey of 554 manufacturing companies in Germany,

Austria and Switzerland show that the most frequently expected advantages of Industry 4.0-related initiatives are more flexible production systems and a reduced time for reacting to changes [14].

1.2 CHANGEABLE AND RECONFIGURABLE PRODUCTION SYSTEMS

The aspired level and form of flexibility of the production systems are determined already during the system design phase. Research focusing on the technical and organizational pre-conditions for flexible production dates back to the 1980s and before [15] and has remained a popular field, as demonstrated by more recent publications [16, 17]. More recently, various forms of manufacturing flexibility are discussed under the overarching concept of changeability (see Appendix A.3 for a brief introduction to the main classes as discussed in literature). This thesis specifically considers a specific type of changeability, namely, Reconfigurable Manufacturing Systems (abbr. RMS). Such production systems consist of modular production machines and resources, such as transport systems, that can be easily rearranged to allow for adjusting of the capabilities and capacity of the production system. Each specific way of arranging the production resources and material flows of a reconfigurable system is called a system configuration [18] and modifications of the configurations referred to as reconfigurations, which can involve including new resources, excluding redundant ones or changing the position of the systems elements with respect to the material flow [19], such as depicted in Figure 2 (left). Therefore, the RMS characteristics allow for reconfiguring the systems and simultaneously imply a broad range of opportunities for combining the modular resources in the system (right). This means that various configurations of RMS can be chosen to realize the same production strategies, i.e. fulfilling specific requirements and objectives for the design and performance of the production system. The variety of designs leading to similar performances was termed the *design space* [20].

Figure 2: Example for production system reconfigurations (left) and design spaces of reconfigurable systems (right)

Hence, the two main characteristics of RMS are (1) that they can be reconfigured and (2) that they make it possible to implement various system configurations with the same resources, creating large design spaces of possible configurations. Yet, the design space can be considered blessing and curse at the same time. On the one hand the many degrees of freedom available to system designers allow for a large number of system configurations

that might be worth considering, enabling the decision-makers to maintain or improve the system performance with each reconfiguration. On the other hand, the design of configurations, their analysis and evaluation are not trivial tasks and generating and assessing large numbers of configurations can be resource-intensive.

1.3 ISSUES OF DESIGNING PRODUCTION SYSTEMS

In addition to the number of potential solutions, various problematic issues were stated with regard to the design processes of production systems and can influence the performance of the resulting solutions [21]. One of them is the holistic perspective needed to design synchronized, well-performing factory (sub)systems, which increases the complexity when compared to designing systems with a more limited scope [21]. Many production machines including their characteristics and parameters have to be taken into account, as well as the features of other resources that represent subsystems of the system under design. Therefore, the sheer number of factors to be considered simultaneously when designing an integrated production system can discourage designers from doing so and put focus on the design of individual subsystems instead. This can lead to detached subsystems that represent acceptable solutions when considered individually, yet stay below the opportunities of more integrated systems. Another factor increasing the complexity of system design is rooted in the interdependencies between the many variables that have to be considered during the design process [21, 22]. Identifying suitable system designs – let alone design strategies – can be challenging if a large number of distinct products with partly varying production process types and production volumes have to be produced, for instance. Hence, trade-offs are to be made between the potentials of an integrated production system and a less complex design process of the subsystems. This method is understandable, considering that the objectives of the production system are often not well-defined at the beginning of the design process [23]. As long as the system's objectives are not anticipated, the efforts to be spent on designing the future system will be chosen cautiously, hence favoring low-effort approaches to design.

1.4 COMPUTATIONAL SUPPORT APPROACHES

To help decision-makers face these difficulties and design well-performing production systems, various approaches have been suggested to solve problems related to system design by using computational support tools. Decision-Support Systems (abbr. DSS) are computer applications to facilitate decision-making in specific problems. Two frequently used techniques in DSS for production system design are simulation and optimization, which differ from each other in the extent of automation.

Simulation deals with the replication of planned or existing processes and system designs [24]. Simplified models of the real system are created and used for analyses with various techniques, such as system dynamics or discrete event simulation [25]. The models therefore allow to automatically analyze the performance of designs in various situations. Simulation models specifically for manufacturing system design have been researched extensively and recent literature reviews suggest that the research in this field is still active [26]. To perform system simulations, the detailed specification of the system's design must be available, which can be created either by human system designers or computational procedures.

Optimization is an approach capable of determining such system specifications. It works on the basis mathematical models of design problems, using objective functions to specify the aspired performance characteristics of the system design, such as minimum cost or maximum output. Based on this specification of the designers' intent, algorithms are used to automatically assign values to the design variables, calculate the design's performance and select the solution(s) that exhibit performances closest to the indicated objective. For the domain of production system design, a significant number of models exist, as well as different computational search strategies for finding the best solutions out of a large number of possible ones. The overviews of the models [23, 27] and search strategies [28, 29] give evidence of the technique's popularity in academic research.

1.5 RESEARCH MOTIVATION

Schotborgh et al. [30] describe the activities and their sequence as typically performed in engineering design, which are visualized in Figure 3. At the outset of the design process the designers have a description of the problem, as well as potential constraints and objectives for the design solutions. Based on this information, a design solution is synthesized and analyzed. Subsequently, the resulting performance of the candidate solution is evaluated. Possible results of this evaluation are that the solution is modified (path a), discarded (path b) or selected (path c), which makes the design become a valid design solution.

Figure 3: Extent of automation inherent in optimization and simulation (based on [30]).

Path a: modification of the design solutions; b: discardment of the design; c: acceptance of the design.

This process model can be used to illustrate the extent of process automation inherent in the simulation and optimization approaches introduced in the previous section. Simulation represents an automation of the analysis process. Benefits of this automation can be time savings with regard to the design process and a detailed account of the designs' performance, however, the latter can be only realized once the system's design is known. This can make simulation a suitable means for efficiently verifying design decisions, but in case the determined system performance is insufficient, the simulation result by itself

does not guide the synthesis of new designs. Moreover, manually developing multiple system designs and then simulating them makes the system designers the bottlenecks in the activity cycle and can be prohibitive in industrial projects, where time and resources are typically limited. In contrast to that, optimization implies an automation of the entire activity sequence and has its own ramifications. An often encountered issue in optimization problems is the time required by the computer to exhaustively generate and evaluate the possible solutions [31]. Moreover, it can be difficult for the decision-makers to express all their preferences in the form of parameters in the objective function [32]. This definition of the 'optimal' performance has a critical effect in optimization, as it implies that only solutions result from the procedure that meet the reference requirements, which might state the designers' intent only inaccurately. This entails that solutions are automatically excluded from consideration because they do not correspond to the analytical definition of optimality, even though these solutions might represent feasible and applicable solutions for the decision-makers. Hence, even though optimization approaches implicitly automate design synthesis and analysis and thereby make it possible to quickly suggest and analyze design solutions, the associated selection procedures imply that the system designers only get to see isolated and potentially unsuitable designs solutions.

In this context, it appears viable to use approaches that combine the efficiency-related advantages of automating the design synthesis and analysis phase with features that support the decision-makers in methodically assessing the design space, i.e. the variety of potential designs. Moreover, the decision-makers should have means to integrate their preferences in a fashion that allows them to avoid the effects of analytical formulations of preference. As applications providing such functionality are hard to find in the context of production system design, there is an opportunity for novel approaches.

1.6 COMPUTATIONAL DESIGN SYNTHESIS

Computational Design Synthesis (abbr. CDS) deals with the automation of design synthesis in the domain of engineering design with the motivation to generate alternative designs [33]. First publications on CDS in mechanical design date back to the early 1990s [34] and CDS in the context of manufacturing was investigated later by Johansson, who determined the potential of design automation for preparing production processes [35]. Nevertheless, explicit mentions in the design of manufacturing and material handling systems could not be found. The objectives of CDS are to achieve effective and efficient design and efficient design processes by automating steps related to design tasks [36] and thus reducing the tedium of human designers [37]. Therefore, the terms of design automation and CDS can be used synonymously. The distinctive feature of CDS-based applications is the automated generation of design solutions, however, the form of executing further and potentially tedious activities such as analysis and evaluation is not exactly defined (see Figure 4). While automating the analysis of solutions can allow for further time savings, the approach to evaluating solutions has to be chosen carefully and suitably for the specific context. An example for evaluating solutions in CDS approaches is to use rules and constraints for automatically discarding inappropriate solution candidates and allowing the decision-makers to evaluate the sufficient ones afterwards [38].

In this way, CDS can be used to create design alternatives that can be subsequently assessed based on the principles of Set-Based Concurrent Engineering (abbr. SBCE). SBCE is a technique associated to the discipline of lean design, which itself is related to the Japanese concept of Lean production. The philosophy behind lean initiatives is mainly to

Figure 4: Extent of automation inherent in computational design synthesis

maximize the value for the stakeholders and eliminate the processes that only increase cost without adding value, also described as waste [39]. With regard to design, waste can be unnecessary and long iterations in the development process, for instance. SBCE attempts to avoid this by developing multiple alternative designs – the so-called set – in parallel [40], as depicted in Figure 5. The various requirements for design solutions, often coming from the different departments involved in the development process, are integrated sequentially. In case the added requirements render specific design alternatives infeasible, these are not considered in the further process. Therefore, SBCE can be distinguished from point-based and conventional concurrent design, where the development of singular designs is executed sequentially (point-based; top in Figure 5) or concurrently (concurrent; center).

In the context of mechanical design, CDS was assessed a high potential for supporting design processes in industrial practice [33]. Furthermore, CDS-based tools that enable the users to operationalize the rationale of SBCE were successfully applied to various design problems, reducing the probability of development process iterations and design lead times compared to point-based design [42, 43]. Industrial application of SBCE also resulted in an improved performance of the designed systems and showed that the design process

Figure 5: Point-based design, concurrent design and set-based concurrent design (based on based on [40, 41])

was more robust concerning specification changes [44]. Therefore CDS and SBCE represent well-researched techniques that appear suitable for addressing the research gap. As a result, research that unravels the implications of support approaches combining automated design and SBCE in the domain of RMS is a worthwhile activity.

1.7 RESEARCH PLAN

The following sections describe the objectives, approach and questions of the research, as well as the context in which it took place.

1.7.1 Research objectives

The overview presented in Figure 6 summarizes the previously introduced topics and their context in this research. Associated to the reconfigurable manufacturing technology is a design space. Design support technologies enable the decision-makers to get insight into the design space and should eventually improve the effectiveness that is demonstrated by the performance of the systems under design. Simultaneously, design support technologies can address the efficiency of design processes. The objectives of the proposed research for supporting the design of production systems are twofold:

The practical objective of this research is to support designers in leveraging the flexibility of reconfigurable manufacturing systems and making the production system design process more efficient and effective.

The scientific objective of this research is to examine how automated design synthesis can be applied to support users in designing configurations of reconfigurable manufacturing systems.

Figure 6: Elements in scope and rationale of the thesis

1.7.2 Research approach and questions

A suitable research approach to reach the illustrated objectives is the Design Research Methodology (abbr. DRM) proposed by Blessing and Chakrabarti [45]. The general objectives of design research are twofold: formulating and validating models and theories about design; and developing and validating tools, methods and knowledge based on these theories, aiming to improve design. The specific objectives of DRM are to help developing an understanding of the design problem and use this understanding to develop means of support. The development of DRM was based on an extensive review of design research

and describes various methods that aim at performing rigorous research by helping to plan and implement design research [45].

Because DRM particularly focuses on developing design support tools and provides a comprehensive research methodology, it is suitable for examining the production system design problem and related solution approaches. The main phases of research projects in the DRM framework are presented in Figure 7: At the outset of the project, the Research Clarification identifies the research objectives and develops plans for reaching them. The first descriptive study aims at creating an understanding of the design problem and identifying criteria for evaluating design support approaches. Based on the understanding of the problem and the criteria for support tools, a support approach is proposed in the prescriptive study. The result of this step can be various forms of support such as software tools, guidelines, or checklists. The suggested support approach is then evaluated in the second descriptive study to determine its applicability, usability and usefulness.

Figure 7: Main phases of the design research methodology (based on [45])

DRM-based projects can involve all of these phases, however often only some of them are covered due to limitations of time or access to critical resources, for example when companies are required for conducting case studies. In each research phase, one of three distinct study types can be performed: review-based research relies on literature study only; comprehensive studies combine literature reviews with empirical results produced by the researcher; initial studies close the project by considering the first few steps of a particular phase without conducting the study to the full extent. Figure 8 presents the seven combinations of these study types – also referred to as research types – that will likely lead to useful research results.

The research presented in this thesis is of research type 6, that is, it consists of a review-based research clarification, a review-based Descriptive Study I, a comprehensive Prescriptive Study and a comprehensive Descriptive study II. While the current chapter demonstrates the results of the research clarification, the research questions to be answered in the other phases are formulated as follows:

- RQ1 – Descriptive study I: What are the requirements for design automation tools to support the configuration design of reconfigurable production systems?
- RQ2 – Prescriptive study: How can the design of reconfigurable production systems be supported by means of automated design synthesis?
- RQ3 – Descriptive study II: What are the characteristics of the suggested approach to design automation?
- RQ4 – Descriptive study II: How applicable and useful is the suggested approach to design automation in practice?

Figure 8: Research types of the design research methodology (based on [45])

Before it can be examined how automated design synthesis can be applied to the design of reconfigurable production systems, a thorough review of the problem is required to identify the relevant objectives for the prospective approach. The resulting requirements for computational support tools represent the answer to RQ1. Based on these requirements, a novel approach is developed which allows for reaching the aspired objectives (RQ2). The evaluation of the support's characteristics (RQ3) and its applicability and usefulness in industrial design practice (RQ4) aim at determining the success and drawbacks of the suggested support tool.

1.7.3 Research context: The Robust PlaNet project

The research presented in this thesis was carried out in the context of the EU-7th framework research project Robust-PlaNet: Shock-robust design of plants and their supply chain networks (referred to as Robust PlaNet; <https://www.robustplanet.eu/>). The participants represented companies and research institutes from four European countries – Hungary, Italy, Germany and the Netherlands – to conduct research of both scientific and industrial interest. The four research institutes were MTA Sztaki – Hungarian Academy of Sciences, Politecnico di Milano (Italy), Karlsruhe Institute of Technology (Germany) and the University of Twente (the Netherlands). The industrial participants were ITConsult Pro (Hungary), OMA (Italy), Knorr-Bremse Hungary, Festo (Germany), Voestalpine Automotive Components (formerly known as Voestalpine Polynorm; the Netherlands), Marposh (Italy) and MCM (Italy). The project started in October 2013 and successfully ended in September 2016. The objective of the Robust PlaNet project were contributions that make production systems more robust against internal and external disturbances [46]. Therefore, the production systems were considered in their context within the respective companies and the supply chains, thus also encompassing the suppliers and customers. Potential means for more robust production were developed and examined for various application domains,

e.g. by providing tools for predictive maintenance for aerospace component machining and for dynamic production planning of automotive braking systems.

The use-case relevant for this thesis assessed opportunities to support decision-makers in the design and operation of assembly systems for automotive body components. Both technology-related aspects illustrated in Figure 6 – technical solutions for realizing a reconfigurable assembly system, as well as support tools for the associated design and management issues – were researched. The results of the use-case were the technical concept of a reconfigurable cellular assembly system and software tools to support the decision-makers in system configuration design, layout planning and production planning. The use-case played an important role regarding the chosen research type, since it allowed the use of the company’s concrete problem specification as a guideline for developing the design support tool proposed in this research. Moreover, the use-case made it possible to evaluate how useful and applicable the tool is, giving insight into its potentials in industrial practice.

1.8 THESIS OUTLINE

Table 1 shows the structure of the thesis. Chapter 2 explains the frame of reference in the domain of production system design and outlines the various issues that play a role when designing reconfigurable systems. Moreover, particularities of design support approaches are introduced. Based on this foundation, the functional framework for support tools is established in Chapter 3, describing the requirements for support tools for RMS design (RQ1). The framework is used to evaluate previous support approaches and characterize their focus. Chapter 4 introduces the prototype tool developed based on the framework, showing an approach for automated design synthesis of RMS (RQ2). This prototype tool is then evaluated regarding its characteristics in Chapter 5 (RQ3). In Chapter 6, its practical applicability and usefulness is evaluated (RQ4). Chapter 7 discusses the industrial impact and opportunities for valorizing the approach. In Chapter 8, the findings are discussed with respect to the initial research motivation, the contributions and limitations of the research are explained and directions for future research are formulated.

Table 1: Overview of the structure, chapters and contents of the thesis

<i>RESEARCH CLARIFICATION</i>	Chapter 1	Introduction, research motivation, objectives and approach
<i>DESCRIPTIVE STUDY I</i>	Chapter 2	Frame of reference: Characteristics of the design problem and computational support tools
	Chapter 3	Functional framework and evaluation of previous support approaches
<i>PRESCRIPTIVE STUDY</i>	Chapter 4	Prototype tool for supporting the exploration of design spaces
<i>DESCRIPTIVE STUDY II</i>	Chapter 5	Evaluation of the tool’s characteristics
	Chapter 6	Evaluation of the tool in practice
	Chapter 7	Valorization
	Chapter 8	Discussion, conclusions and recommendations

Part II

DESCRIPTIVE STUDY I

The following two chapters present the frame of reference and understanding of the problem. Chapter 2 highlights the relevant properties of production system design problems and computational support tools. Building on this background Chapter 3 determines the functional features that support approaches would require in order to address the design problem properties. Furthermore, existing support tools are reviewed and their capability to provide the required functionality is discussed. The result of the two chapters is a solid theoretical basis for developing the design support tool.

FRAME OF REFERENCE

This chapter presents the related topics and literature from the associated fields of research, dealing with design and development of production systems in general (Section 2.1), the specifics of reconfigurable production systems (Section 2.2) and computational design support (Section 2.3).

2.1 PRODUCTION SYSTEMS

To build understanding of the problem, the following sections introduce characteristics of production systems and the design approaches relevant to this research, as well as managerial issues of production system design.

2.1.1 *Design characteristics*

SYSTEM COMPONENTS At the beginning of Chapter 1, production systems were functionally defined as systems that convert specific inputs into products and can be considered on various levels of aggregation. At the most detailed level of consideration, the resources constituted in production systems can be categorized as follows [3, 47, 48]:

- **Production machines:** Production machines are equipment for executing specific production processes. Different degrees of automation of production machines can be distinguished (1) manually operated, (2) semi-automated, or (3) fully automated. Manual operations require human workers throughout the process; semi-automated processes require human workers only during parts of the operation, such as for feeding the raw material; fully automated processes can be executed without requiring regular human intervention.
- **Tools, fixtures and other related hardware:** Tools perform the processing operations in conjunction with the machine and fixtures can be required to hold the product during the processing. Other related hardware includes dockings for temporarily unused tools, for example.
- **Material handling systems:** Such systems provide for all movements of the products between the individual production processes. These systems can take many shapes, such as manually drawn carts, conveyors, forklift trucks or automated guided vehicles. In addition to these instances of material handling systems, warehouses and buffers can be considered a special form of material handling system that provide raw materials, semi-finished goods and final products in a stationary manner.
- **Industrial robots:** Robots are a special class of production machines that are versatile and can be used for many different purposes. Besides executing production operations they can be used for material handling tasks, for instance.

- Human workers: Workers can perform various tasks in production systems, such as directly executing production processes, indirectly producing by controlling the machines or handling materials. Workers also set-up and maintain the systems or program the robots.
- Computer systems: Computers are used for operational planning and automated control of manufacturing processes, also referred to as computer-aided manufacturing.

Figure 9 illustrates various production system components that cooperatively perform production processes in two production subsystems. The material handling system transports the semi-finished goods to the next operation. Both production cells perform production processes and hence represent separate production systems, however, also the overall resulting system represents a production system. Hence, at lower levels of consideration the production system is characterized by the used hardware, whereas when choosing a higher level, the combined, systemic features and chain of production and material handling processes characterize the system. Moreover, even though the components can be discriminated from a physical point of view, specific system components can fulfill multiple functions in the production system. In the illustrated example, a single robot could support production in subsystem 1, the material handling and production in subsystem 2. Therefore, the involved components, their combined functionality and the system boundaries are relevant features that describe the design of production systems.

Figure 9: Example for delimiting production systems in various ways

PRODUCT SCOPE Another fundamental technical characteristic of production system design is the product types to be produced with the system. Two extreme forms can be identified: general purpose systems (also called universal systems) that can produce multiple products, and product-focused systems that are designed exclusively to produce a single product (called dedicated systems) [4]. An intermediate form are systems designed for product families, producing multiple products with similar, however not identical processing requirements. Since systems can be considered on various levels the product scope has to be referred to the respective levels, i.e. plant, department, workstation. Deciding to produce multiple products with the same system typically affects the system's design and performance. If multiple products are meant to be produced on the same workstation and require different machine tools, the system's technical design must provide opportunities to perform tool changes. Moreover, the implications of tool changes for the system's performance must be considered during system design, since each tool change represents a loss

of capacity that could be used productively otherwise. To keep the capacity losses due to such changeovers at a reasonable level, batch sizes are defined for each product, describing the quantity of each product that needs to be produced before changing the tools and starting the production of another product. Figure 10 shows the influence of the batch sizes on the system's performance: on the one hand, larger batches can reduce changer-over related capacity losses; on the other hand, smaller batches sizes can reduce the lead-time of the products from the customer order until delivery of the products. As can be seen in the figure, the lead-time difference of the two batch sizes can be significant for the same customer demand volume. Therefore, the implications for the system's performance have to be assessed and compared to the customer requirements in addition to finding technical solutions for producing multiple products.

Figure 10: Change-over related capacity losses

PRODUCTION PROCESS TYPES AND SEQUENCE An additional aspect to differentiate production systems is the type of operations performed on the products and parts. One way of classifying the main types of production processes looks as follows (based on [48]):

- **Casting processes:** Processes that give materials an initial form, for instance by casting liquid metal into a mold or rolling it to coils.
- **Forming/shaping:** Processes that affect the shape of products, including bulk deformation, sheet-metal forming and polymer processing.
- **Machining:** Processes that reduce the volume of workpieces, e.g. by applying milling, drilling or turning.
- **Joining/assembly:** Processes that combine two or more parts to a product, for instance by welding, screwing, gluing or riveting.
- **Finishing:** Treatment processes such as honing, sand blasting or painting are used to improve specific product properties.
- **Micro- / Nanofabrication:** Processing technologies that can be used to create products with very high precision, on micro- or nano-level, such as etching or lithography.

These process types have different logistical implications. For instance, finishing operations are characterized by the same ratio of inputs to output components. Forming and

Figure 11: Different logistical implications of the production process types

machining processes are special, since they can both imply 1-to-1 ratios of input and output parts (see left in Figure 11; e.g. bulk deformation of a part, turning of a shaft), as well as 1-to-n ratios (center; e.g. piercing parts from a blank, cutting spiral wire to compression springs). In contrast to that, joining operations require two or more parts as input and combine them to a whole (right). These parts often need to go through a longer sequence of preparatory operations before they are ready to be assembled. The resulting interdependencies and sequence of assembling the final product, its subassemblies and components can be expressed in the form of assembly trees (see Figure 12). In the example, the nodes on component level (component 1 to 5) are independent, however the remaining nodes 6 to 9 represent subassemblies that require these subordinate components. Eventually, these dependencies have to be taken into account when planning the material flows in the system and influence their complexity. For designing systems at the department level, often

Figure 12: Example of an assembly tree (based on [49])

multiple products with various process and assembly sequences need to be integrated and make system design a complex activity.

PROCESS EXECUTION AND AUTOMATION Another aspect characterizing the design of production systems is the level of process automation. As described in Section 1.1, the equipment for automated execution of production processes has become increasingly affordable. Decision-makers have to decide whether manual or automated execution of each process in the system is more suitable and cost-efficient. Even though automation frequently implies higher initial investments than required for manual processes, the cost-per-product can be significantly reduced when compared to manual labor due to faster and more reliable process execution, enabling a positive return on investment if sufficient quantities of products are produced [50]. Hence, automation can be an option to make production, material handling, transport and storage processes more efficient. Another important aspect that needs to be considered is the geometric features and tolerances of the products to be produced [51]. By performing processes with lower variability compared to the manual execution, automation can be a means to produce at higher quality levels [50]. Yet, the benefits depend on the exact specifications of the problem, i.e. how much faster or more reliable the automated process performs and how much investment is required for automation. Manual processing and handling of products can be more viable due to difficulties with regard to automating specific processes (e.g. handling of non-rigid parts), the high motoric and sensory flexibility of people or low product quantities that make profitable automation improbable. Therefore, it has to be considered that the specific level of automation affects the design, cost and performance of production systems significantly, whether the processes are considered at the level of the workstation, line, department or the entire factory.

SYSTEM ORGANIZATION TYPES By considering the production system on the department level various system organization types can be observed. Chryssolouris defines five basic organizational principles of executing production processes, as depicted in Figure 13 [23]:

- **Job-shop:** The job-shop describes a system type, where the manufacturing processes are grouped by process type. Products that require various process types are transported between the process groups.
- **Flow line:** Flow lines are production lines in which the products follow a defined sequence of operations, thus it is particularly suited for products requiring similar process types and sequences.
- **Cellular system:** In cellular systems, the resources for executing the production processes for a specific part (family) are grouped to production cells. Inside those cells, the products can follow different process sequences or require different processes.
- **Project-shop:** In a project-shop organization the required processing resources are transported to and from the stationary products, which can be advantageous for bulky or heavy products.
- **Continuous system:** In continuous systems the products also follow a defined sequence of steps. This organization type is only listed for completeness since it is intended for bulk products, liquids or gases and not for discrete parts or products.

Figure 13: Organization principles of manufacturing systems

For the production of discrete products, the organization types 1 to 4 are applicable. Production systems based on the job-shop, flow line and cellular principle can involve an automated processing of these products inside the individual production systems, such as in the process group 'AA' in system type 1. The benefits of choosing a cellular system is that the processes are performed without requiring separately coordinated material handling between the processes in the system, therefore executing the processes faster. Additionally to that, the flow line principle provides better accessibility of the individual operations and thereby provides more opportunities for supplying the individual operations with raw materials. The improved accessibility is also a major reason for the project-shop, where the operations are executed on the stationary product. Hence, a characteristic difference between the concepts is the number of production processes that have to be coordinated separately and the positions and movement of products and processes relative to each other. Moreover, the degree of concatenation differs between these organization types. The programming, failure or maintenance of an individual machine in an automated flow line or cellular system can force the entire line/system to be inactive [11], whereas the same event in a job-shop or project-shop concerns only the respective operation, leaving the other processes unaffected. In factories, combinations of the organizational principles are common [23]. One motivation for combining the principles is partly heterogeneous and partly similar processes of multiple products, such as explained by the example presented in Appendix A.4. The organization types can also express different degrees of specialization. Job-shop systems represent organization types focused on the separate tasks, whereas flow lines are more aligned to the specification of the product(s). Cellular systems are hybrid system types with focus on both products and tasks. As a result, the organization types embedding and connecting the individual processes are a characteristic design feature of production systems.

PROCESS REQUIREMENT AND CAPACITY Another important factor and one of the strategic design variables in production system design is the process requirement of the product

portfolio and the capacity offered by each of the involved systems [4]. Figure 14 visualizes the material flows of three products that are produced on three production systems. Each production system has local buffers for temporary storage of the products. Since delayed provision of raw materials leads to a waste of processing capacity and reduction of the output, the buffers on the input side of the systems need to be filled, which is performed by the transportation system. At the same time, the buffers on the output side need to be emptied once they are full. In case any buffer's capacity is insufficient, the products need to be buffered in the main warehouse, which is indicated by the dashed lines and can cause additional efforts. Hence, the capacity utilization of the production, transportation and storage systems are mutually dependent. Careful planning is required when deciding which resource should represent the system's bottleneck. In the illustrated example, Production system 1/PS1 is the bottleneck since the available capacity is fully utilized. Even though the other systems have spare capacity, the specific material flows make it impossible to achieve increase the production volumes since all products have to go through PS1. Therefore, the material flows create interdependencies that require to keep a holistic perspective during system design, to consciously decide on the allocation of capacities and encompass all involved systems, i.e. considering the flow within individual lines or cells and also the flows between multiple lines or departments.

Figure 14: Example for capacity requirement and resource selection in the context of material flows

Consequently, the technical design of production systems is characterized by: (1) which hardware is involved; (2) how many and which types of subsystems are involved; (3) how many and which products are produced; (4) which process types and operation sequences can be performed; (5) which degree of automation was chosen for the production and handling processes; (6) which approach to system organization is chosen; (7) how much capacity the resulting system provides.

2.1.2 General engineering design approaches

The following section introduces design and development theories from general engineering design applicable to production systems. They represent only a share of the various theories that were established so far. Readers who would like to get an overview of engineering design theories should refer to Adam's explanations [52] or to Evbuomwan [53] and Tomiyama and Schotborgh [54] for more detailed discussions of their commonalities and differences.

ENGINEERING DESIGN The field of engineering design examines systematic methods for solving technical problems to allow for successful product development [55]. Since the discipline focuses on design of technical systems in general, it is also applicable to production system design. A well-known approach to engineering design was proposed by Pahl et al. [55]. Their theory distinguishes the four iterative phases of task clarification, conceptual design, embodiment design and detailed design (see Figure 15, left). At the start, a clear description of the problem and the objectives of the design task is developed, before creating suitable solution concepts. The form of implementing the concepts is then determined by choosing an embodiment and specifying the designs' details. Even though advances in the field of engineering design methodologies have been made since the approach was first described in 1977, the basic rationale of the theory is still considered valid [55]. Another design process model was suggested by Schotborgh et al. and was already used to explain the extent of design automation in Sections 1.5 and 1.6. It distinguishes the main activities in the design process [30]. The main phases passed through for each design are referred to as synthesis, analysis and evaluation and are linked as presented in Figure 15 (right). In the synthesis phase, an idea for solving a particular problem is discretized by proposing a candidate solution (steps a and b). During the analysis phase, the performance of the solution is quantified (c). Then the solution is assessed based on its resulting design and performance characteristics (d). In case this evaluation indicates that the candidate solution is not good enough, the designer can modify and optimize it (e) or generate a new candidate (f); otherwise the candidate is accepted and becomes a design solution (g). Iterating through this loop represents what is typically known as trial-and-error, in which

Figure 15: Left: Engineering design methodology (based on [52, 55]); right: Generic design activities (based on [30])

designers use their experience to develop suitable solutions for the respective problem.

SYSTEMS ENGINEERING Systems Engineering (abbr. SE) represents another branch of science dealing with the design and development of technical systems. SE is an interdisciplinary approach that promotes defining the customer needs and required system functionality at early stages of the system development process [56]. Furthermore, SE stimulates to put focus on the documentation of requirements, design synthesis and system validation [56]. SE provides a rather flexible set of concepts and techniques that can be helpful for designing systems, yet without prescribing certain approaches and in which sequence to apply them [57]. Therefore, SE can be considered a set of loosely coupled techniques that share common objectives, rather than an exactly defined theory or approach [56]. The commonalities of these techniques are (1) top-down design processes that consider the systems holistically; (2) interdisciplinary development of the systems; (3) clear specification of requirements for the systems; (4) and an emphasis on considering the systems' entire lifecycles [56]. SE addresses organizational issues in development projects and started to emerge as an own field of study when support was sought for the design of complex systems in space and defense programs. Today, SE is relevant for general engineering systems and hence also for the design of production systems. Maier and Rechtin, for instance, point out the importance of the intersection between the lifecycles of production systems and products (see Figure 16) [58]. Since the lifecycles of production systems are often longer than the ones of products, the production engineers have to anticipate future products and thus look further into the future than product architects, making SE in production systems a challenging task. As explained in the previous section, the performance of production systems depends on more than just the systems for processing the products, since also the systems for transport and storage of materials have to be considered. In SE, System-of-Systems (abbr. SoS) are a separate research field that encompasses the challenges and techniques related to the integral consideration of multiple systems involved simultaneously. Although various definitions of SoS exist, their shared basic characteristics are that they combine systems that are separately defined and operated, but work together towards a common goal [59]. Thus,

Figure 16: Intersecting product and production system development waterfalls (based on [58])

SoS concepts are relevant for production system design, since production systems often involve lower-level subsystems (for production, material handling for transportation and storage) and the requirements for these systems come from various organizational units within the company, e.g. finance for investment and cost, logistics for stock levels or sales for lead-times and product cost [10, 60].

SUMMARY In summary, the presented theories address different views of design. The approach suggested by Pahl et al. identifies separate phases of design, ranging from undefined, over conceptual and embodiment design to the completely specified product or system. The approach of Schotborgh et al. focuses on the types of design activities – synthesis, analysis, evaluation and optimization – and their sequence in design processes. Therefore, the iterative activity cycle can be passed through for each of the major phases of design, i.e. for conceptual, embodiment and detailed design. SE provides concepts that support the definition of the problem structure, objectives and focus of the design processes in the organizational context.

2.1.3 *Dedicated theories of production system design*

Additionally, there are also design theories that exclusively focus on production systems. Some similarities of the general and production-specific design theories can be observed, since the latter originated from product development [61]. Discussions of general production system design theory are presented in [21, 61, 62]. A distinction among the design theories can be made, based on whether they consider the design of systems holistically or as decomposed design processes, focusing on solving partial problems [4, 20]. In the decomposed approaches, decisions are made separately with regard to adjacent domains (see Figure 17, left). For each domain, the performance requirements are mapped to values of the system design variables. In this manner the resource requirements, the resource layout, the material flows and buffer capacities are determined separately [23]. Yet, it was argued that such isolated consideration of subproblems might be suitable only for systems with little interaction between the system components; holistic approaches should be taken to avoid local optima [63]. Therefore, design frameworks should ideally consider a large number of aspects of the production system design problem in an integrated way. An example of such an approach is the sequence of activities presented in Figure 17 (right), suggested by Bellgran and Säfsten [4]. In the main design phase (design and evaluation), the preparation of design and the design specification are distinguished. Next to that, the planning of the design process itself and the implementation of the production systems are described as separate phases of the development framework.

Despite their different form, the two approaches presented also overlap in some aspects. In particular, the mapping of performance requirements and system design variables of the decomposed approach is an activity that takes place implicitly during the design specification phase of the holistic approach. Yet, the holistic approach does not distinguish the various design subproblems and rather seeks to address them in an integrated manner. Moreover, Chryssolouris' decomposed approach reflects a general classification of problems solved in literature and is therefore primarily a reference for scholars of solution methods for the particular subproblems. Its only commonality with the engineering design approaches is that the mapping of design variables and requirements is mentioned as part of the activities in the design theory of Pahl et al.. In contrast to that, the more general perspective of the holistic theory presented by Bellgran and Säfsten aims at provid-

Figure 17: Left: Decomposed production system design (based on [23]); Right: Holistic production system design (based on [4])

ing industrial decision-makers a flexible approach to production system design that can be adapted to the unique situation of companies in practice [4]. As it describes partly similar stages (conceptual and detailed design) and activities (design and evaluation), the theory exhibits a significant overlap with both engineering design theories presented.

2.1.4 Managerial issues of production system design

Various issues that can affect the design process and its management have to be considered when applying either design approach practice.

JUSTIFICATION OF SYSTEM DESIGNS One of these issues is that production system designs commonly need to be justified based on both technical and economic criteria [64]. An overview of research describing various technical and economic performance indicators as well as performance measurement systems for production systems is given by [65]. The approach taken to measure the performance plays an important role for achieving strategic objectives and keeping non-value adding activities in flexible production systems to a minimum [66, 67]. While value-adding and non-value-adding activities can be distinguished by defining the concepts from a technical point of view – e.g., value-adding means that activities/systems transform the parts and products – the economic valuation is more complex due to large degrees of freedom of attributing cost. One aspect creating room for interpretation is determining which cost types should be involved in the evaluation, such as cost for system hard- and software, their operational cost or capital costs of required investments. After that, the second potentially debatable aspect is how to attribute the cost to processes and products over a defined time horizon [68]. Companies often use multiple internal and external performance measurement schemes in parallel and thereby attribute cost in multiple ways [68]. Since each approach can result in a unique characterization of the systems' performance, the use of different approaches to the economic valuation of system designs can lead to differing – potentially also contradicting – judgments of the design's viability. Hence, it is essential to value system designs with a performance measurement approach that is agreed on by all system stakeholders and which consistently and transparently reflects the causality of the systems' design, related activities and cost.

REQUIREMENTS IN GREENFIELD AND BROWNFIELD PROJECTS Another aspect to be considered during the design of production systems are the requirements from the environment for the system. A general distinction of greenfield and brownfield design projects is presented in Figure 18 [69]. Greenfield refers to designing systems without having to take into account constraints from existing infrastructure, which is commonly the case when the system should be placed in a new factory or part of a building that is not yet built. This makes it possible to create the technical design of the system first and align the design of the building subsequently to guarantee that the system and the building are compatible. This is required, since industrial production facilities often require special infrastructure, such as systems for supplying electricity, compressed air or water treatment. Also, the payload of machines, cranes and other production infrastructure must not exceed the specific physical bearing capacities of the floors and roofs in the facilities. Since the building of production facilities is capital-intensive, the lifecycles of the buildings are typically longer than those of the production systems they accommodate [70] so that the required, significant investments can be utilized as long as possible. As a result, designing production systems to be operated in facilities that are already established – referred to as brownfield design – typically involves requirements that stem from the existing infrastructure, such as limited area for realizing future system designs due to space that is occupied by neighboring systems. Such constraints can take numerous shapes and thereby limit the degrees of freedom for production system design significantly compared to greenfield design.

Figure 18: Interdependencies between facility design and production system configurations (based on [18, 69, 70])

SYSTEM-OF-SYSTEMS ENGINEERING The organization of the production systems and constituent subsystems also to be considered during their design. The design of production systems on a low level of consideration, such as workstations, is a task that can be fulfilled by a single engineer. Yet, increasing the level of aggregation (e.g. considering entire departments) often results in SoS, where the constituent systems are designed and operated by multiple decision-makers. The resulting SoS display various degrees of coordination: (1) directed SoS consist of subsystems that are centrally planned and managed to fulfill the main purpose of the SoS; (2) virtual SoS are not managed centrally and thus can exist without the decision-makers being aware of the SoS; (3) collaborative SoS are planned centrally (such as the directed ones), but cannot exert power over the subsystems and hence rely on voluntary collaboration of the subsystems; (4) acknowledged SoS have designated objectives and a manager of the SoS, but the subsystems have independent ownership, objectives, funding and unplanned collaborative mechanisms [71]. In their case study at enterprise level, Muller and Andersen observed that the production system was

a virtual SoS, despite their expectation to find a directed or acknowledged SoS [60]. This highlights that a pre-requisite to holistic SoS design is that the decision-makers are SoS aware [72]. Chikán found that even though production and logistics are often described from an integrated perspective in literature, in practice they seem to be managed in a less integrated way [73]. Therefore, typical characteristics of SoS that can be observed with regard to production systems particularly at higher levels of consideration are as follows: there is integral ownership, i.e. all production and material-handling subsystems are required to produce the products and contribute to the overall cost and performance; there is a lack of overview of the involved systems; the systems have emerging properties, i.e. the actual system performance is a result of the interplay between the various subsystems and can deviate from the predictions made during design; and many stakeholders make decisions locally [60]. Therefore, specifically during brownfield design projects for re-designing specific subsystems (e.g. individual production cells) it is challenging, but also important to consider the system holistically and design solutions not based on an isolated viewpoint. Due to the interdependencies of the various subsystems, their design should take place in a coordinated fashion.

2.2 RECONFIGURABLE MANUFACTURING SYSTEMS

Reconfigurable manufacturing systems (RMS) have additional features that distinguish them from general production systems. Therefore, this section introduces to the relevant characteristics, design approaches and associated managerial issues.

2.2.1 *Design characteristics*

Even though RMS make use of equipment components similar to general production systems, it is not possible to generally define which are the reconfigurable elements and how they are supposed to be reconfigured/changed [74]. Since reconfigurability aims at enabling re-use of equipment to increase the equipment lifecycle value [75], the decision which components should be re-used strongly depends on the economic value of the system component. Due to the context-specific and vastly differing ranges of the value of system components, the reconfigurable and static system components need to be defined individually for each case.

CONFIGURATION DESIGN PROBLEM Nevertheless, the technical design of RMS allows to ascribe further characteristics to the design problem, additionally to the general aspects introduced in Section 2.1.1. Technical enablers that make systems reconfigurable were identified as follows (factors (1) to (5) [18]; factor (6) [61]):

1. Modularity requires the system to be composed of individual modules, instead of a rigid whole.
2. Integrability is given if the modules can be easily integrated to form a whole, which requires compatible physical, functional and control interfaces.
3. Customizability refers to the quality of the total system to only provide the exact functionality and capacity needed in the specific moment to avoid unnecessary cost. Simultaneously, a customized control architecture is required to obtain adaptable system designs.

4. Convertibility describes the capability to adjust the system's functionality in short time, considering activities such as the changing of tools, fixtures and control programs.
5. Diagnosability aims at enabling the operators to quickly detect quality problems during production. This allows to quickly increase the production volumes during the ramp-up phase after the system has been reconfigured.
6. Mobility describes the capability to change the location where production process are executed and can be interpreted with regard to the individual modules of the system or the system as a whole.

In the RMS design process concrete embodiments need to be tailored to the specific products and processes to fulfill these functional requirements. Therefore, the first step in configuration design problems is to determine suitable solution topologies and configuration components, as illustrated in Figure 19 [76]. After these aspects are defined, components need to be selected and allocated to eventually obtain solutions that fulfill the prevalent requirements and constraints [76–78]. Hence, RMS can be described more concretely than general production system design, since the specific structure of the systems is required for making them reconfigurable.

Figure 19: Characteristics of system configuration design problems (based on [76, 77])

RECONFIGURATION TYPES Reconfigurable system architectures enable to perform two forms of reconfigurations, physical and logical ones [74]. Physically reconfiguring the system means to change the production equipment such as machines, tools, fixtures, inspection machines and material handling systems or their arrangement. Therefore, physical reconfigurations can be observed at the lower levels of production systems, i.e. workstation or production cell. Logical reconfiguration takes place by adding, removing, or changing the modules of the system thereby changing the functionality or the capacity and can lead to re-routing of products. Hence, this reconfiguration type can be mainly observed at the system or higher levels. Reconfiguring the system by removing a production machine from one production cell and installing it in another one can involve a re-routing of products and therefore represents both reconfiguration types simultaneously.

INNOVATIVENESS OF CONFIGURATION DESIGN Another characteristic of RMS is the possible range of solutions and their innovativeness. Gero distinguishes three major types of design: routine design, innovative design and creative design [79]. Routine design is characterized by a well-defined space of potential designs, which means that all design variables and possible value ranges are known. Nevertheless, only a subrange of the possible ranges is produced and evaluated (Figure 20, left, black circle). Innovative design also takes place within a well-defined design space, yet the produced designs represent broader

Figure 20: Possible and produced spaces of routine (left), innovative (center) and creative design (left) (based on [79])

ranges of the design variables than considered in routine design (center, grey circle). The resulting designs have a familiar structure (e.g. consist of the same components), but are novel in appearance. Lastly, in creative design new design variables are determined, resulting in new solution types, hence extending the state-space (right, white area). Since reconfigurability requires the definition of the topology and components of design solutions, the main design variables are known. Therefore, configuration design yields routine or innovative designs.

2.2.2 Design method

RMS and their characteristics have been discussed intensively for almost twenty years. However, many authors stated that design theories are missing and the general manufacturing system design theories do not sufficiently take into account the implications of reconfigurability [74, 80, 81]. The RMS design framework suggested by Andersen presumably represents the first attempt to generically describe the process, resulting from an extensive review of prescriptive RMS design literature [74].

Figure 21: Generic RMS design method (based on [74])

The method starts with planning the design process and clarifying the task. Subsequently, the basic design of the system is specified, deciding on the variety of products to be produced on the system and the degree, type and level of reconfigurability [74]. In this phase, the first context-specific design decisions are made. In the following advanced design phase, the design of the modules takes place. Also, the physical and logical aspects of the system are decided at this stage, among others. Following this, the system is implemented and production starts. After the operational phase the system can be reconfigured, which involves an iteration back to the previous stages.

Similarities exist among the proposed framework and the previously presented theories for general engineering and production system design, since it distinguishes between the phases of basic design and advanced design, which is a resemblance with the phased approach proposed by Pahl et al. [55]. The iterative element in the phases of basic and advanced design is also described in the approach suggested by Schotborgh [30]. Moreover, the inclusion of the initial planning and clarification explicitly refers to the approach proposed by Bellgran [74]. The RMS design method synthesizes the main aspects of the reviewed prescriptive literature and focuses on the reconfigurable design concept in the basic design phase and definition of the modules and interfaces in the advanced design phase. However, the resulting solution topologies, components and interfaces only represent partial results of the configuration design activity. As shown in the previous section, the selection and allocation (abbr. S&A) of the modular components is an integral phase in configuration design, which is however not explicitly reflected in the proposed RMS design approach. Hence, due to their importance the S&A steps should be also reflected in methods for RMS design that consider the entire system lifecycle. The resulting modified model of the design process is visualized in Figure 22, showing the initial design of the RMS and the first reconfiguration. When completing the initial advanced design phase, the detailed design of the modules, interfaces and control system is known (see asterisk). To maintain the reconfigurability of the system, it is essential that these characteristics remain throughout multiple reconfigurations. Therefore, when reconfiguring the system, the basic and advanced design activities primarily apply to designing new modules that can be integrated into the system. In contrast to that, the S&A steps form the core of the configuration design task and are repeatedly performed throughout the initial design phase and the reconfigurations.

Figure 22: Activities and phases in the lifecycle of reconfigurable production systems (based on [4, 74, 76])

2.2.3 Managerial issues of RMS design

UNCERTAINTY Many technical design aspects of production systems are implicitly related to the customers' demand and preferences. For instance, the viability of automating a specific process depends on the number of products to be produced in a defined period, which in turn depends on the expected demand volume and delivery dates. In reality, the actual demand can deviate even if customers forecast their future demands in advance

[82]. Design decisions are often based on forecasts or other data sources, such as company-internal estimations, both implying high uncertainty. This uncertainty relates to the fact that specific environmental conditions lie beyond the control of the decision-maker and is described as external uncertainty [83]. Additionally, also internal uncertainties can affect the design process that relate to factors that can be controlled but are often unintendedly influencing and aggravating the decision making, such as vague information and imprecise judgements [83]. Hence, when designing production systems, internal uncertainties can lead to misjudging the requirements or objectives for the system design, leading to system designs and behaviors that are not suitable or intended. External uncertainty plays a role in the design of manufacturing mostly in the form of the unknown future demand of product types and volumes [61], which can render the most suitable system designs for specific situations unsuitable in case unexpected developments emerge. Hence, uncertainty can make it difficult to provide a solid basis for designing the system and simultaneously exacerbate the possibility to identify realistic performance objectives for the system under design. As a result, the lack of control of these variables should be taken into account during the RMS design procedure and stimulate the system designers and stakeholders to explicitly state requirements and objectives based on shared assumptions.

COMBINATORIAL NATURE Another challenge related to the S&A steps in configuration design is the large quantity of solutions that stems from the number of opportunities for combining the components to integrated designs. For instance, the design of cellular production systems (explained in more detail in Section 4) requires to group production machines to one or multiple production cells, which is exemplified in Figure 23 (left). With each additional machine to be grouped, the number of combinations for grouping – also called partitions – can grow to a level that makes it impossible for designers to enumerate the design options, as shown in Figure 23 (right) by illustrating the growth of possible solutions/partitions (y-axis) when considering n machines for partitioning (x-axis). The growth of the number of partitions is characterized by the Stirling number of the second kind (formula and theory explained by [84]), which grows faster than the exponential function based on a growth factor of two. Once a set of machines was selected and grouped, a high number of solutions can be possible also for allocating the products or production

Figure 23: Example for partitioning three elements (left); Comparison of Stirling combinations with exponential growth.

steps to the machine groups. As a result, the definition of suitable components and solution topologies is only the first challenging step of RMS design; assessing a large number of potential alternatives when designing configurations – required for exploiting the system’s flexibility – is another challenge the decision-makers face during each reconfiguration.

RECONFIGURABILITY The S&A steps involve selecting the most suitable modular resources and allocate them to obtain system configurations that fulfill the decision-makers’ preferences. The aspired objective of reconfiguring the system is to increase the system’s lifecycle value by using the opportunities for adapting the systems with respect to changing stakeholder needs and re-using the system components [75]. Figure 24 compares RMS to flexible manufacturing systems (FMS) and dedicated systems (DMS) with regard to system cost and capacity (left), as well as the variety and degree of customization of the products to be produced (right). The cost-capacity classification (left) shows the intended effects of RMS: an improved scalability of capacity in comparison to dedicated systems; and a lower gradient of cost and capacity compared to flexible manufacturing systems. These differences seem to be reflected also by the classification based on the product variety and production volume (right). Customizing low production volumes requires less capacity than mass customization, hence flexible systems appear the better choice to produce a high variety of products with low volumes. Simultaneously, dedicated systems are better suited for mass production, as they are dedicated to specific products or product portfolios. Even though both characterizations attribute overlapping regions of RMS with both flexible and dedicated systems, the structure of RMS should make them efficiently scalable for producing a medium variety of products at medium volumes [74, 85]. Designing configurations

Figure 24: Flexible, reconfigurable and dedicated manufacturing systems; Left: Capacity and system cost (based on [18]); Right: Production volume and product variety (based on [16])

of RMS in real situations requires decision-makers to consider their product portfolio and concretely operationalize the term of medium product variety by choosing specific products. Also, it is necessary to define the medium volumes that serve as basis for planning. As shown above, these volumes are commonly uncertain, making it difficult to realistically categorize them as high, medium and low volume products. Similar difficulties may apply in case the production volumes are more certain, however the differences between them are incremental. Hence, the selection of products at the outset of the design process involves subjective evaluations that significantly influence the required system design characteristics (such as process types), the resulting number of potential system designs and their performances.

Moreover, the external uncertainty can influence the exact course of the graph representing RMS depicted in the left part of Figure 24. Hence, the reconfigurations actually performed can deviate from the initial planning due to developments that are difficult to anticipate. Therefore, one of the main benefits of RMS is the multiple alternative designs – the design space – that make it possible to achieve the same objective in various ways (e.g. producing the same set of products in multiple ways). Yet, the design space can also enable decision-makers to achieve different objectives, i.e. move the focus to system designs with different performances. This can be beneficial in case customers change their preferences, e.g. when customers preferred low-cost products and change their priority to short product lead-times. In case low-cost products are demanded, the acquisition of the modular resources and their integration during configuration design can focus on obtaining low-cost structures; in contrast to that, the expectation of shorter lead-times can be fulfilled to a certain extent by switching the focus of system designs to the speed of the resources, flows of materials and aggregate time for production, transportation and storage. Therefore, RMS allow for a reactive approach. Nevertheless, the decision-makers need design strategies able to reproduce those options and consider the various objectives to be capable of utilizing the technical opportunities and reacting in different ways.

REQUIREMENTS DURING RECONFIGURATION An additional aspect applying to RMS is the effect of reconfiguration on the overall productivity, which is illustrated in Figure 25. Changing from the current to the future configuration requires to halt the production. Depending on the size and technical features of the system modules, reconfiguration can take a significant time that depends on the (dis)similarity of the current and future configuration. Next to that, also the efforts such as the number of required workers performing the reconfiguration can differ and should be considered when evaluating potential system designs. Consequently, the implications of reconfiguring the system can be decisive for the viability of adopting a specific system configuration and thereby represents a potential domain for requirements for future system configurations.

Figure 25: Effect of reconfiguration on productivity (based on [18])

2.3 COMPUTATIONAL DESIGN SUPPORT

This section presents various forms and relevant aspects of computational design support tools.

2.3.1 Types and characteristics of support tools

As stated in Section 1.4, computer applications can be classified based on their purpose. The concept of Decision Support Systems (DSS) refers to the general capability of software tools to make decision processes more effective and allowing the users to control the decision-making [86]. One specific type of DSS are Knowledge-Based Systems (KBS), which are computer programs capable of solving problems, using program routines that apply human expertise formalized in so called knowledge bases [87]. Another term often used to describe such applications is Expert Systems which is therefore synonymous to KBS [86, 88].

A characteristic property of DSS that can be used to distinguish applications is the degree of interaction with the user. Hättenschwiler proposes a classification that distinguishes passive, active and cooperative DSS [89, 90]. Passive decision support means that the software is not capable of suggesting solutions for a problem. Regarding the problem of production system design, simulation tools constitute examples of passive DSS as their main purpose is to analyze solutions. Active support software is capable of proposing solutions, so that optimization tools can be considered representative for this type. Lastly, cooperative systems allow their users to modify, complete or refine solutions that were proposed by the system, involving an iterative process [89]. The three forms of support imply various functionalities that have to be collectively provided by the Graphical User Interfaces (abbr. GUI), routines and data structures, which represent the main components of the DSS.

With regard to active and cooperative DSS, the way of formulating preference information concerning the design and performance of solutions can be used to characterize such systems. Four different forms of articulating preference information can be distinguished [91]: (1) the preference is not articulated; (2) the information describing the decision-makers preference is specified before solving the problem (termed *a priori*; e.g. optimization); (3) the preference is iteratively defined after assessing (sets of) solutions (progressive); (4) the preference is stated only after generation of non-dominated solutions (*a posteriori*). In this

Figure 26: Design preference expressed in terms of requirements (left) and objectives (right)

context, the preference can express two things: solution characteristics that are aspired and those that are not wanted. Solution characteristics that are (not) wanted can be formulated in terms of requirements, such as presented in Figure 26 (left). In the example, the requirements for the two parameters p_1/p_2 are that solutions should not have values of p_1 that are lower than x and simultaneously p_2 lower than y . Different from that, objectives only describe the aspired qualities of solutions (see right). The example shows the solutions when the objectives are high values of p_1 and p_2 simultaneously. Hence, objectives are considered relative preferences that are only vaguely defined, whereas requirements represent concretely defined boundaries used to evaluate solutions. Both can be formulated at various moments and in ways that lead to the same design solution(s).

2.3.2 Design space exploration

An activity that is particularly important with regard to active and cooperative DSS that are capable of suggesting multiple solutions is Design Space Exploration (abbr. DSE; also referred to as Tradespace Exploration [32, 92]). As suggested by the name, the purpose of DSE is to explore the design space, methodically compare the various designs before selecting a specific design and aims for discovering new ideas or re-assuring the suitability of existing ideas [93]. DSE as computational support approach has been applied in various design problems, such as design of computer systems [94] and mechanical systems [95]. Also in the context of designing changeable mechatronic systems the approach was applied and has proven to be beneficial for successful system development [92]. DSE requires suitable features for representing and visualizing solutions to enable the users to evaluate the suggested solutions. Figure 27 illustrates the possible value ranges of two design variables d_1 and d_2 (left) and the system performances p_1 and p_2 (right).

Each dot represents one design solution, described by its design and performance attributes. As indicated by the arrows, features such as GUIs for interactive visualization make it possible to assess solutions with specific values for p_1 and p_2 and thereby find

Figure 27: Visual exploration of the design space (based on [95])

combinations of d_1 and d_2 that fulfill the users' requirements, and vice versa. In this way, the various properties can be related to each other and make it possible to assess individual solutions and compare multiple solutions in the design space. Hence, such features allow the users to explore the design space and make it possible to get an understanding of the underlying relationships of the different variables of the solutions before specifying a final set of preferences and constraints [32]. Relevant evaluation criteria for DSE approaches are the quality of the resulting design solution [94] and the efficiency of the process for finding solutions that reflect the preferences of users [96].

2.3.3 Managerial issues of design automation tools

An important characteristic of design support tools is their economic viability, described by the efforts required for their development and application, the efforts saved by their use and the effects on the quality of the design solutions. Figure 28 describes the temporal sequence of the activities involved in the development, deployment and utilization of design automation tools.

Figure 28: Development activities for design automation tools (based on [97])

EFFORTS OF SUPPORT TOOL DEVELOPMENT As the first step of developing design automation tools, the design problem to be solved is decomposed for gaining an overview and determining system boundaries for the software [97]. Afterwards, the problem-solving knowledge is acquired, e.g. by interviewing the experts and actors in the problem's environment. The next step is to model the design problem and determine synthesis procedures capable of automating the generation of alternative solutions. The resulting design automation model can be implemented in a software application that makes it operational. Before the resulting design automation tool can be applied, it has to be integrated into the application environment, which can require trainings of the prospective users or adaptations of the processes that form the preceding and following steps of the tool's usage. In case the tool should be repeatedly applied, also its maintenance is a major topic and cost-driver to be considered [98] and can be required to update the knowledge models and procedures, for instance.

BENEFITS OF SUPPORT TOOLS Detrimental to the efforts caused, the potential improvements of design automation tools mostly concern two main areas. Firstly, the software tool can lead to improvements with regard to the efficiency of the design process, e.g. reduced

cost of staff, reduced process durations, higher productivity or higher learning rates. Secondly, design automation tools can allow for developing better system designs, e.g. lead to production systems with reduced production lead-times, reduced stock, lower cost, higher productivity or more flexibility. While the benefits of design automation tools for the efficiency of the design process might be anticipated more accurately, it is often difficult to estimate their effects on the quality of design in advance. This visualizes a dilemma for decision-makers who have to decide about investing into support technologies, since without developing the support tool and a related model, the design-related potential for improvements may remain unknown and hence unused. However, after developing and applying such tools and models it may turn out that the actual improvement potential is too small to compensate for the efforts spent. Hence, the main stakeholders and beneficiaries should ideally have a notion of the potential benefits of the tools or at least an awareness of the expected cost of providing the support, so that the financial risk becomes manageable and can be mitigated.

LIFECYCLE CONSIDERATION OF SUPPORT TOOLS. The total effects of the support technology are visualized in Figure 29 and can be considered in analogy to the operative result of reconfigurable manufacturing systems. Initial investments are required for developing the reconfigurable system architecture and component interfaces. Additional costs are associated to each reconfiguration and – if necessary – to the development or procurement of additional system modules. In case the evolution in the longer term is uncertain – which can be assumed to apply to most of the cases in which RMS concepts are employed – the cost savings are difficult to forecast and investments can be difficult to justify. The timings, as well as types of expenses, benefits and risks can apply identically to the design support technology. Initially, the support technology needs to be developed and each reconfiguration creates another opportunity for applying it. Nevertheless, also the required behavior of the support technology is influenced by uncertainty. Hence, the support tools should be easily adaptable to make it possible to apply them under changed problem specifications or, alternatively, their development, deployment and application must be sufficiently fast and cost-efficient so that their utilization can be justified for individual configuration design decisions.

Figure 29: Profit/loss during lifecycles of RMS and support tools (based on [99])

AUTOMATING RMS DESIGN: FUNCTIONAL REQUIREMENTS AND APPROACHES

This chapter highlights the link between RMS design and features required by computational tools for supporting it and thus presents the answer to the first research question: *What are the requirements for design automation tools to support the configuration design of reconfigurable production systems?*. Section 3.1 discusses the relation between the characteristics of the RMS design problem and the required support tool features. The results are summarized in a functional framework for RMS design support tools, which is used in Section 3.2 to compare various existing approaches. Lastly, Section 3.3 concludes the chapter and the descriptive study.

3.1 FUNCTIONAL REQUIREMENTS FOR DESIGN AUTOMATION TOOLS

First, the implications of design automation and the most suitable RMS design process phases are discussed. After that, the support tools features required to address the identified RMS design issues are identified and summarized into a functional framework for design support tools.

3.1.1 Suitable RMS design phases for design automation

The three major phases in which different details of the RMS design are specified are basic design, advanced design and the selection and allocation (S&A) of components. While computational tools could theoretically address each of those phases, approaches that involve an automation of design are most suited for the S&A steps. Since the design of production systems and RMS is context-specific, the required production technologies, tolerances and other specifics of the production system depend on the products to be produced and also the reconfigurable features of the RMS, which are not generally defined [4, 100]. Hence, the production system and its reconfigurable features need to be determined specifically for each problem case, which may involve creative design activities to achieve the aspired functionality, such as forming unique modules by physically joining distinct functions of production hardware components. So far, design automation is not suited for such creative design tasks [36]. The development of design automation procedures requires to formalize the design knowledge and corresponding models in knowledge bases before the knowledge can be automatically applied by algorithmic procedures. Therefore it is difficult to automate the basic and advanced design phases since this requires to anticipate the results of a potentially ill-defined design task. In contrast to that the combinatorial S&A steps can be automated. Once concrete models of the system design variables – represented by the RMS topologies, components and their interfaces – can be created, sufficient system design

information is available to formalize procedures that automate the selection and allocation of modules.

3.1.2 *RMS-related requirements for design automation tools*

As explained in Section 2.3.2, computational tools need to provide a few basic features to support users in design decisions. The algorithms, user interfaces and data structures of support tools have to be tuned to the particularities of the considered design problem to yield a harmonized functionality and effectively support the designers. The particular issues of RMS design presented in the previous chapter promote further features that should be provided by support tools in the S&A steps. These are presented in the following.

SHOWING THE VARIETY OF SYSTEM CONFIGURATIONS IN THE DESIGN SPACE The chosen concept to make the production system technically reconfigurable might only specify some of the basic production system design characteristics. For instance, the definition of a reconfigurable production cell architecture with modular machines does not necessarily specify how many of those cells should be installed, how many machines to be installed in each cell and where the processing of the products should take place. In the same fashion, the definition of the RMS architecture can leave freedom for allocating the processes and capacities, as well as for the resulting system organization and material flows. Therefore, the selection and allocation of modules can still involve to decide about various remaining system design aspects. Important factors that influence the degrees of freedom available in the S&A steps is the level at which the system is considered, since it determines which subsystems, potential subsystem components and interactions at all levels have to be taken into account during the design. Section 2.2.3 shows that the number of possible configurations can become quite large even if there are only few properties for combinatorial design. Moreover, only small differences in the design of configurations can lead to a large difference of their performance due to the systemic effects resulting from the integration of multiple systems, as illustrated in Section 2.1. Hence, tools for supporting the S&A steps of RMS require routines that are capable of generating system configurations that show the variety of the RMS design characteristics with regard to the main degrees of freedom and specifically for the considered context.

ENSURING THE APPLICABILITY OF SYSTEM CONFIGURATIONS An important element in all design process models presented in the previous chapter is formed by the requirements of the decision-makers, which characterize the outcome of the design process, embodied by the design solutions. The integration of the requirements during the design process is essential for obtaining design solutions which are applicable solutions for the considered design problem from the perspective of the decision-makers. Various requirements are potentially relevant with regard to RMS. For instance, the set of products the prospective system configuration should produce is a requirement with significant effects on the resulting system design. This aspect is commonly determined and operationalized at an early moment of the design process. Further requirements can concern the basic design characteristics, as well as the technical and economic aspects of the integral SoS and its constituent subsystems. Considering again the example of the reconfigurable cell architecture, specific values or value ranges can be required for the number of cells to be installed, the number of machines, the system utilization and so forth. Requirements can also stem from brownfield situations, previous configurations, as well as the allocation of

products or processes and characteristics of material flows (e.g. centered vs. distributed flows). The specific combination of requirements determines what the design space looks like, i.e. which parts of the design space are feasible and which are not. Therefore, support tools must provide opportunities to integrate the situation-specific constraints for the decisive variables, e.g. the products, the system's design, performance and reconfiguration. Yet, the users must also be able to change their requirements, since the interdependency of the design and performance variables of system configurations makes it difficult to predict the effects of particular requirements on other characteristics of the system design. It is possible that a requirement involuntarily eliminates opportunities from the design space that represent interesting solutions. When room for negotiating particular requirements exists (e.g. products to be produced) or the exact value of the requirement cannot be specified with certainty (e.g. required production volumes), various combinations of these requirements are possible, but each combination can have its own implications for the design space. In this situation, the users of support tools should be able to experimentally test those combinations and identify if and how they affect the design space, the design solutions and the aspired characteristics which can be obtained. As a result, support tools should provide features for integrating the requirements so that users can consider negotiable and uncertain requirements related to the system's design or performance and customize and assess the related design spaces.

ENABLING USERS TO DETERMINE THE QUALITY OF SYSTEM CONFIGURATIONS Once tools are capable of generating customized design spaces, it is essential for the decision-makers to be able to assess the generated designs with regard to their technical and economic characteristics, compare them to one another and eventually select superior solutions. The selection of system designs has to be justified based on the various criteria for comparing the solutions. As explained in Section 2.3.2, all systems design and performance characteristics that can be used to express requirements can be also used to compare and express objectives. Moreover, the system aspects to be compared should reflect the evaluation criteria of the various design stakeholders and therefore should be agreed on by all decision-makers to allow for holistically and objectively comparing the designs. Next to the technical design and operational performance of system configurations, the indicators should also quantify the implications of reconfiguring the system to allow for trading off the expected performance improvements of system configurations against the effort required to implement them. Similarly, the sensitivity of system configurations with regard to uncertain factors such as product demand should be factored in to allow for a quantification of various scenarios the system designs can face. Consequently, the support tool features must enable the users to navigate the customized design spaces and eventually allow to select the most suitable candidates based on the technical and economic RMS characteristics with regard to system design, operation and reconfiguration.

ENSURING THE SPEED OF AUTOMATED SYNTHESIS, ANALYSIS, EVALUATION AND SELECTION As shown in Section 1.1, novel production technology and system paradigms became prevalent because they allowed for more efficient production processes. The striving for efficient processes is also common in other areas of business, such as product and system design. In fact, a substantial part of the approaches to engineering and production system design directly proclaim objectives related to the efficiency of design processes [4, 23, 55]. Hence, also design support tools have to be efficiently applicable, considering the efforts for using the tools in each configuration design project. In this context, the efficiency

is strongly influenced by the speed of passing through the distinct design process phases and the uncertain or ill-defined requirements. To formulate requirements in various ways and examine the implications for the resulting designs, all design activities – generation of solutions, analysis, evaluation and selection – must be executable sufficiently quick. As the term ‘sufficient’ indicates it is difficult to specify a general limit for the aggregated duration of application, since the available time depends on the situational context. A reference value for such duration could be a typical working day. If multiple iterations with varying problem instances and requirements should be performed on a single day, decision-makers should be able to perform one complete iteration in less than two hours. When allocating this total duration to the individual activities, a few factors have to be taken into account. One important consideration is that the design synthesis procedure must be capable of generating a variety of solutions, that is the various context-specific design options. Similarly, the analysis of system designs has to be adjusted to the allowed maximum duration by employing an analysis approach that is sufficiently accurate and efficient. For evaluating solutions, both the users’ requirements and objectives have to be integrated in reasonable time. In these manual steps, features of the support tool should support and help to speed up the decision-makers’ orientation in the design space, i.e. helping to exclude irrelevant solutions and compare interesting solutions that match the users’ objective. Also the eventual selection of viable system designs based on the users preferences should be supported by software features to help speeding up the process. Hence, tools for supporting the S&A steps of RMS design should allow for generating, customizing and navigating the design spaces efficiently.

3.1.3 Functional framework for tools automating RMS design

The RMS-related requirements for design support tool features can be summarized and put into relation with the overall objectives of design support. The resulting model is presented in Figure 30 and explained in the following.

Figure 30: Functional framework for design automation tools for the S&A steps of RMS design

MAIN OBJECTIVES OF DESIGN SUPPORT As Wu states, methodologies for manufacturing system design should be supported by computational tools to be effectively applied [101]. Therefore, one of the main objectives of such tools can be seen in their capability to increase the effectiveness of the design process. The testimony of the effectiveness is the match of the resulting system design and its performance with regard to the decision-makers' requirements and objectives. The second main objective and measure of the design process is its efficiency, measured by the required resources such as people, tools, information and time and its performance in comparison to the number of relevant alternatives assessed. In this context, both the application of the support tools and their life-cycle related efforts such as development and maintenance have to be considered, as explained in Sections 2.3.3 and 3.1.2. Hence, without providing for an acceptable efficiency of the support tools, the effectiveness-related benefits of such applications are likely to remain unutilized.

RMS-RELATED OBJECTIVES OF SUPPORT FEATURES For reaching these main objectives, the RMS-related support tool features identified in the previous section represent the technical enablers. Support tools should make the variety of design opportunities accessible for evaluation by the decision-makers. Various possible solutions should be considered concurrently, which implies that the number of solutions is another relevant factor. The synthesis and analysis of solutions has to be performed at an acceptable speed to allow for generating many solutions that represent various designs. Yet also the pace of evaluating and selecting solutions is relevant. The decision-makers have to be able to integrate their requirements for design solutions to ensure that the resulting designs are applicable, i.e. fulfill the minimum expectations. Similarly, opportunities for integrating objectives are required to allow the decision-makers to find system designs exhibiting the aspired qualitative characteristics. Lastly, another important objective is to provide customizable support tools, allowing to assess various related problem statements without requiring major modifications that cause additional efforts.

SUPPORT FEATURES For achieving these objectives, support tools require specific features, which cannot be concretely specified for two reasons. Firstly, the support tool must integrate functionalities for synthesizing, analyzing, evaluating and selecting solutions, yet, various forms of implementation are possible to achieve these functionalities. For instance, requirements could be formulated progressively, requiring a different approach than formulating them a posteriori. Secondly, the exact form of reconfigurability and related component models used to automate the S&A steps are specific for the situational context of the considered problem. The characteristics used by the decision-makers to evaluate solutions might differ between application domains (e.g. layouts, logical reconfiguration, ...) and the tools' features for supporting the evaluation should address these aspects by providing suitable procedures and GUIs. Hence, the intended modes of stating preferences (none, a priori, progressive, a posteriori), the activities (requirements integration, objectives integration, comparison, sensitivity assessment, selection) and the RMS system design properties have to be considered in the situational context to determine the required form of the features to achieve the identified functional objectives. Since the synthesis and analysis of solutions should be automated, the objective of fast application promotes the idea of automating also preparatory data analyses. While an interactive DSS approach is required for iteratively customizing the design space, the degree of automation for integrating the requirements and objectives has to be defined in the specific context.

3.2 SUPPORT TOOLS FOR AUTOMATED PRODUCTION SYSTEM DESIGN

A number of approaches for automating the design of production systems are described in literature. Therefore, this section evaluates to which these tools incorporate the features of the developed functional framework.

SCOPE OF REVIEWED APPROACHES A literature search was performed for support applications that involve design automation in the configuration design steps of RMS to find previously suggested approaches and evaluate them based on the framework. Yet, literature describing tools with such functionality could not be found. Reasons for that could be that RMS started to become a relevant research concept only during the first decade of the 21st century (see [74] for publication statistics) and design methodologies and support approaches still lack consolidation. Hence, the literature survey focuses on support tools involving design automation for various production system types. A potential problem associated to this strategy is that other production system types might not have the same pre-conditions as RMS configuration design, that is a pre-defined design task that makes possible a broad range of alternatives. Yet, all problems solved by the evaluated support tools can be understood as configuration problems since the applied solution procedures presuppose defined solution topologies, requirements and objectives of the resulting systems, as shown in the following. This implies that the considered design problems are treated as configuration design tasks and hence resemble the nature of RMS configuration design problems. As a result, the state-of-the-art analysis considers tools that automate the design of various production system types, such as flexible manufacturing systems or conventional assembly lines.

DESIGN SUPPORT TOOLS The main categories for comparing the DSS are listed in the headers of the columns in Table 2. The author-column provides information about the authors' names and year of publication. The remaining columns are split into two groups that describe the properties of the design support tools and those of the considered design problems. Regarding the support tool properties, the type of the support tool and the user interfaces are categorized, as well as the characteristics of the support tools with regard to the users' opportunities for specifying requirements and objectives. Moreover, the tools' features for considering uncertainty are described and the number and variety of resulting solutions. The last evaluated support tool property is whether the tools allow the users to perform integral comparisons of the generated solutions. The design problem properties are described by the design problem type and the main design variables taken into account.

The starting points for the DSS proposed by Chan et al. is the quantitative strategic objectives for production system design with regard to operational and financial performances [102]. After that, an expert system proposes and modifies the system design until the objectives are met. Following that an analytical-hierarchy approach selects the best system design, using neural networks to assign selection criteria weights and fuzzy logic to define the membership functions.

Chan and Chan propose another DSS specifically dedicated to the design of production lines for printed-circuit boards [103]. It employs an expert system to determine the minimum number of machines required for executing the production processes and then performs a simulation of the configuration to determine whether the resulting capacity utilization lies between the required boundaries. In case the required utilization is not achieved, the number of machine instances is changed and an iteration is started.

The DSS proposed by Chtourou et al. also combines an expert system with simulation, allowing the users to interact with the tool through a GUI [104]. Based on an initial solution and user-specific requirements for the result, the tool automatically changes and simulates system configurations to identify opportunities that allow for achieving the performance requirements. As a result, the most effective design changes are suggested to the users.

Khan et al. describe a KBS for determining the number of machining centers, choosing material handling systems and the layout [105]. The system obtains the users' requirements by posing multiple questions before synthesizing a matching solution and performing a financial analysis. They validate their approach by performing multiple case studies and comparing the resulting solutions with the ones developed by expert system designers. Malakooti and Kumar developed a KBS for multi-objective assembly line balancing, which includes decisions on the number of workstations and the allocation of work to those stations [106]. Initially, the users specify their requirements and constraints, based on which the system determines suitable candidates. Eventually, the solution identified by the automated multi-criteria decision analysis are shown so that one solution can be selected manually.

Mellichamp et al. propose a KBS for flexible manufacturing system design that generates and analyzes multiple options for configuring the production system and automatically compares the resulting designs to the initially defined objectives and requirements [107]. After the system has identified solutions that correspond to the users preferences, the users can select the most suitable option.

Michalos et al. developed a KBS that automatically determines the number of resources required for multiple processes in an assembly line and subsequently configures the line with various equipment components [108]. The selection of particular line configurations is based on multi-criteria decision analysis and manual comparison by the users.

Rao and Gu propose a KBS that interacts with a neural network to develop system configurations for cellular manufacturing [109]. The system proposes ways for clustering the processing equipment to production cells and determines the allocation of parts. Once a solution is capable of producing all required products, it is accepted.

DSS TYPE AND USER-INTERFACES All of the assessed DSS represent active approaches that propose solutions based on specific data input, yet without giving users significant opportunities for interaction. Little interaction between the procedures and the user is also reflected by the description of GUIs, which are illustrated in only three of the approaches. Of the remaining approaches, only one provides a description of the GUI and four do not mention interfaces at all. Obviously, the early approaches (such as Mellichamp et al. and Malakooti and Kumar) were developed in the 1990s, when opportunities for visualization and interaction between the support tools and the users was rudimentary compared to today. Yet, the lacking description of the interfaces of comparably recent approaches (Khan et al.; Chan and Chan) shows that the tool developers mostly focus on automating the entire design activity range, i.e. automating synthesis, analysis, evaluation and selection, barely providing opportunities for interacting with the users.

INTEGRATION OF REQUIREMENTS The resulting style of interaction also affects the moment when users can specify requirements for the solutions, which is performed a priori in all approaches. The way of integrating these requirements differs between the support tools. For instance, the approaches of Mellichamp et al. and Chan et al. implicitly implement requirements in their proposed models. In contrast to that the support tool of Khan et

Table 2: Problem aspects and support features of DSS for production system design

AUTHORS	DSS TYPE	USER INTERFACE	INTEGRATION OF REQUIREMENTS	<i>Properties of the design support tools</i>				<i>Properties of the considered design problems</i>	
				INTEGRATION OF OBJECTIVES	CONSIDERATION OF UNCERTAINTY	NUMBER OF RESULTING SOLUTIONS	INTEGRAL COMPARISON OF MULTIPLE SOLUTIONS	EXAMINED DESIGN PROBLEM	MAIN DESIGN VARIABLES
Chan et al. 2000 [102]	Active	Not shown, only described	A priori; defined by the model; not customizable	A priori; defined by the fuzzy criteria; not customizable due to AHP	not considered	Single solution	No	Not concretely defined	Not defined, depends on model
Chan and Chan 2005 [103]	Active	Not described	A priori; defined by the model; not customizable	Requirement satisfaction is only objective	not considered	Single solution	No	Dedicated to production line for printed-circuit boards	Number of machines for pre-determined process sequence
Chtourou et al. 2005 [104]	Active	Input of requirements; Output of suggested changes	A priori; through GUI; partly customizable	Requirement satisfaction is only objective	Stochastic demand model	Single solution	No	Not concretely defined	Not defined, depends on model
Khan et al. 2011 [105]	Active	Not described	A priori; defined by the questions; customizable	Requirement satisfaction is only objective	not considered	Single solution	No	Not concretely defined	Not defined, depends on model
Malakooti and Kumar 1996 [106]	Active	Dialogue-based/ textual in- and output	A priori; through GUI; customizable	A priori/ A posteriori; initial multi-criteria decision analysis; final solution selected manually	not considered	Multiple solutions	Mainly performance KPIs, only one design characteristic	Not concretely defined	Not defined, depends on model
Mellichamp et al. 1990 [107]	Active	Not described	A priori; defined by the model; customizable	A priori/ A posteriori; initial multi objective comparison; final solution selected manually	not considered	Multiple solutions	Mainly design characteristics, only two performance KPIs	Not concretely defined	Not defined, depends on model
Michalos et al. 2015 [108]	Active	Input of requirements; Output of solutions	A priori; through GUI and model; partly customizable	A priori/ A posteriori; initial definition of criteria weights; final solution selected manually	not considered	Multiple solutions	List of design and performance indicators	Automotive body assembly line	Location, number and types of machines for assembly line design
Rao and Gu 1995 [109]	Active	Not described	A priori; defined by the model and by input parameter values	Requirement satisfaction represents only objective	not considered	Single solution	No	Cellular manufacturing	Number of cells and machines; product allocation

al. obtains part of the requirements based on questionnaires on runtime of the application. Chtourou et al. and Michalos et al. provide graphical user-interfaces that allow the users to specify requirements for some system design variables. Requirements that are inherent in the model cannot be easily customized, whereas entering requirements through GUIs on runtime provides opportunities for modifying them. However, even the approaches that enable the users to customize the requirements do so only for specific system design properties, such as performance (e.g. Chan and Chan) or design variables (e.g. Rao and Gu, Michalos et al.). None of the approaches makes it possible to integrate requirements for all relevant system design and performance properties. Hence, in case requirements apply to particular system properties (such as shop-floor space requirements, investment costs or other undesirable solution characteristics) and cannot be applied in the support tools, it is possible that the solutions determined by the tools are infeasible.

NUMBER, VARIETY AND COMPARISON OF SOLUTIONS Even though the approaches are designed to automatically generate, analyze and evaluate a variety of designs – hence perform an automated design space exploration – five out of the eight tools only lead to a single solution. Hence, potentially interesting alternative designs are hidden by the automated selection procedures. Among the approaches that produce multiple solutions, the way of comparing the solutions are distinctive. In the approach by Malakooti and Kumar, the solutions are presented in a table that compares them based on a single design property (number of workstations) and a few performance characteristics (cycle time, buffer size, production rate, total cost). The solutions of the DSS proposed by Mellichamp et al. are also presented in a table, making it possible to compare various resources for performing the process in relation to few performances (utilization, queue length). However, it remains unclear whether the table is part of the support tool or was designed only for the publication. Only the approach suggested by Michalos et al. shows an interface for visualizing the solutions' design properties (details of configurations) and performances (availability, investment cost, reutilization, solution ranking). The approach allows to compare multiple solutions based on one performance parameter at a time, yet it seems to be difficult to involve the solutions' design characteristics into the comparison or compare design and performance parameters simultaneously.

INTEGRATION OF OBJECTIVES The visibility of solution properties in the various support tools is also important with regard to formulating objectives. Since the approaches of Michalos et al., Mellichamp et al. and Malakooti and Kumar can produce more than a single solution, they allow for a manual integration of objectives based on the generated candidates and limited to the respective solution properties. This makes it possible that system design and performance characteristics are not contained, which are actually required by the user for evaluating the designs. The other principles applied in the support tools range from requirement satisfaction as only objective (e.g. Chan and Chan, Chtourou et al.) to multi-criteria decision analysis (e.g. Michalos et al.) and combinations of fuzzy logic and neural networks (Chan et al.).

UNCERTAINTY, ITERATIONS AND SPEED With regard to the influence of uncertainty, only the support tool suggested by Chtourou et al. involves a model feature for considering stochastic product demand. None of the other approaches provides features for considering the uncertainty in any form. As a result, an important assumption immanent to the vast majority of the tools is that all parameters are known with certainty and there-

fore a solution can be easily identified. This deterministic and absolute perception is also reflected by the fact that none of the approaches foresees or mentions iterative application of the tools as means to let the users assess the solutions resulting from a slightly varied problem specification. Moreover, also the time requirement for applying the tools to specific problems is indicated only for the approach of Michalos et al.. This makes it difficult to assess whether iterations can be performed in an efficient manner, even though the tools do not foresee iterations by design.

DESIGN PROBLEMS AND VARIABLES A difference between the approaches can be observed also on the subjects of the design problems and variables in focus. Michalos et al., Rao and Gu, as well as Chan and Chan present support tools for clearly defined problems, yet the other approaches are mostly presented generically and demonstrated by applying them in case studies. The applicability of the general approaches and quality of their solutions cannot be determined, unless they are verified and validated based on specific problems preferably from practice, such as performed by Khan et al.. In this context, it might prove difficult for industrial problem owners to transfer the rationale of the generic design automation approaches to the specifics of their problem domain. With regard to the design variables considered by the support tools it should be noted that limiting consideration to specific parts of the production system can make it difficult to use the approaches in related, but slightly different problems. For instance, the approach suggested by Michalos et al. considers a large number of product-related parameters to configure the assembly line, yet the scope is limited to a single product. Consideration of additional products – even with similar process sequences – is difficult. Hence, support tools should make it possible to examine related design problems and, in case the problems have to be repeatedly solved, anticipate potential evolutions of these problems. Also the DSS proposed by Mellichamp et al. can be used to illustrate the importance of this factor. The model considers various hardware configurations of a flexible production system for multiple products. Particularly when facing uncertain demand information, it might be interesting for the decision-makers to assess the implications of setting up multiple systems and distributing the production volumes to multiple systems. This adds another dimension to the problem, since next to the design of the system also the allocation of products and potential material handling between the cells must be considered. Based on an extensive review of production system design methodologies, Ruffini concludes that prescriptive methods often lack scope (areas the design theories cover), range (environment for which they provide valid solution), detail or a combination of these factors and argues that those are likely causes for their rare application [20]. The examples made with regard to the approaches of Michalos et al. and Mellichamp et al. illustrate how these factors can analogously apply to design automation approaches. Therefore, the effort required for developing support tools that are only capable of addressing a fraction of the relevant problem formulations can detain decision-makers from applying them in production system design. Simultaneously this consideration could provide an explanation for the sparse practical application of approaches involving design automation, which is indicated by empirical research in production system design (see Appendix A.5). As a result, it appears advisable to develop models that are capable of supporting the users in related problem definitions, rather than providing models whose capabilities are narrowly limited.

SUMMARY With regard to the findings presented above, the following conclusions can be drawn. Even though the tools represent rather sophisticated approaches to automated

design problem solving, they exhibit shortcomings with respect to modeling the problem environment they are supposed to be used in. One of these drawbacks is that all of the presented approaches involve stating the requirements and objectives a priori, if they allow for customization at all. At the same time, holistic comparison of the solutions with respect to the relevant design and performance characteristics of the system is not possible with most of the tools, either because only individual solutions are generated or only few solution properties are described. Merely the approach of Michalos et al. represents an exception that provides features for supporting the comparison and evaluation of multiple solution characteristics. In combination, the limited opportunities for changing the requirements and evaluating the designs can cause the resulting solutions to deviate from designers' intent and thereby render the design support approaches ineffective. Moreover, the vast majority of approaches does not allow for considering uncertainty and assumes all objectives and requirements to be clearly known to the users at the outset of the process. This represents an implicit assumption that stands in contrast to the external and internal uncertainty explained in Section 2.2.3 and the issues highlighted in Sections 1.4 and 2.1.4. In none of the presented approaches the efforts related to development and modifications are quantified, making it difficult to get an impression of the economic viability of developing customized models.

These findings also appear relevant for the general debate of design automation in the domain of production systems. Yin et al. emphasize the importance of intelligent human-machine integration when automating design to generate benefits by combining the massive computation and logical deduction abilities of computers and the experience, intuition and creativity of humans [110]. Yet they note that 'unfortunately, although design theory and methodology have been quite well developed in the last few decades, they are still lagging behind the development of advanced manufacturing technologies and systems' [110]. An important problem – and potential explanation in this context – found by other researchers with regard to computational tools for production system design, was the limited orientation of the proposed design support tools at the actual design process [101]. Hence, despite making progress in the development of sophisticated computational techniques, the findings of the assessed DSS indicate the support tools' principles of application – i.e. the interfaces between users and computers in the problem solving procedure – have not changed significantly over the years. An interesting question relating to this was posed thirty years ago by Mellichamp and Wahab, who proposed one of the tools presented above: 'What is the 'optimum' machine/human interface for such a system (design support system; author's note)? Is it realistic to build a system that ultimately removes the human designer from the loop, or should the system be structured in such a way as to challenge and motivate the innovative, creative potential of the human designer?' [111]. In general, the presented support tools are of a rather prescriptive nature and only require human decision-makers to feed input data, pick one of multiple solutions based on narrow comparisons in exceptional cases and implement the emitted design solutions. The DSS of Michalos et al. enables the users to get limited insight into the design space and therefore represents an approach with slightly more user-involvement. Yet, the other approaches almost entirely remove the system designers from the solution of the actual design problem. This makes it an interesting endeavor to determine opportunities and implications of approaches to design support that involve the users to a greater extent, such as the tool combining design automation and DSE proposed in the next chapter.

3.3 CONCLUDING REMARKS

Chapter 2 presents the frame of reference of the research by outlining the design characteristics, theories and managerial issues for both general production systems and reconfigurable ones. Also, criteria for distinguishing design support tools and their link to design space exploration are explained. Based on this foundation, this chapter answers the first research question (*What are the requirements for design automation tools to support the configuration design of reconfigurable production systems?*) and shows that an important prerequisite for DSE – an automatable design task – is fulfilled only in the selection and allocation steps of RMS design. The various required features of support tools for automated design of RMS are explained and integrated in a functional framework. Most importantly, the objectives are efficient and effective design processes, which are both strongly related to the ability to consider solutions for related problem definitions and integrate individualized requirements and objectives. A literature survey of tools suggested for supporting production system design shows that many approaches make it difficult to generate and assess various designs, due to rigid forms of integrating the requirements and objectives in the design automation procedures, as well as limited opportunities for holistic comparison of system designs. The findings indicate that support approaches that adhere to the framework bear the potential for interesting additions to the research field.

Part III

PRESCRIPTIVE STUDY

*Learn to fail with pride - and do so fast and cleanly.
Maximise trial and error - by mastering the error part.*

— Nassim Taleb Nicholas

This chapter describes the computational tool developed to address the support tool objectives explained in the previous chapter, thus providing an answer to the second research question: *How can the design of reconfigurable production systems be supported by means of automated design synthesis?*.

THE DSET – A TOOL FOR AUTOMATED, SET-BASED DESIGN OF RMS

The starting point of this thesis was the motivation to avoid analytical formulations of preference and enable the decision-makers to assess various, feasible design solutions for their problem. The computational support tool developed in the course of this research – referred to as the Design Space Exploration Tool (abbr. DSET, pronounced di-set) – combines the concepts of design automation, design space exploration and set-based design. It automatically generates various production system configurations and has features for DSE that enable the users to assess and evaluate those solutions. By allowing the users to integrate their requirements, it actualizes the rationale of SBCE. Section 4.1 outlines the empirical problem context and delimits the scope of the tool. Then, Section 4.2 introduces the DSET's features and rationale. Section 4.3 explains how the developed DSET aims to fulfill the requirements for support tools presented in the previous chapter. After that the relevant implications and assumptions of the approach are discussed in Section 4.4, before concluding the chapter in Section 4.5.

4.1 PROBLEM DEFINITION AND SCOPE OF APPLICATION

The DSET was developed based on the system design problem of the company in the focus of the Robust PlaNet project use-case, a supplier of assembly modules for car bodies. Traditionally, the company operated assembly cells that could be hardly changed once they were set up on the shop-floor. Driven by the uncertainty of future orders and the potential benefits of reconfigurable systems, a reconfigurable system architecture was developed. This architecture, the aspired benefits and the related challenges for system designers are explained in Section 4.1.1. Afterwards, Section 4.1.2 delimits the scope of the DSET in the problem context.

4.1.1 *Problem definition*

The developed production system architecture is a cellular assembly system with (multiple) automated production cells and reconfigurable cell components, referred to as a reconfigurable cellular assembly system. The concept of cellular manufacturing is considered a derivative of the group technology concept, which has been subject of studies for almost a century until now [112]. Research on cellular manufacturing has been performed over the last fifty years [112]. As explained in Section 2.1.1, the term 'cellular' refers to the characteristic logical structure of the factory shop-floor, which consists of production cells. The considered cellular assembly system aims at generating economies of scale by automating the execution of the processes in the production cells. The concept of reconfigurability is actualized by some of the components of the production cells that can be changed. Fig-

Figure 31 depicts the architecture of the reconfigurable cells, which consist of essential and reconfigurable modules. In this context, the term ‘essential’ implies that the modules are required for operation, whereas the number and types of the reconfigurable components can vary. Inside the assembly cells, the processes are executed as follows: the products and parts are inserted into the cell via the turntable (1), so that the 7-axis robot (2) can pick them up and bring them to the Functional Assembly Groups (abbr. FAGs; 3) that are placed on either side of the robot’s rail, next to the FAG-specific machine tools (4). After processing on one or multiple FAGs, the products are put onto the conveyor belt (5) that moves them out of the cell. Automation of the process from input to output is controlled by the control unit (6). The FAGs are classes of assembly operations that make use of similar equipment, such as spot welding, resistance joining, gluing or riveting. Therefore, they contain the equipment required for the specific class of operations (e.g. the power source of a welding machine) and the installations for attaching the product-specific machine tools and fixtures. The machine tools in the cells can be changed when starting the production of a batch of another product and therefore can be required multiple times per day.

Figure 31: Reconfigurable cell architecture

The reconfigurations can be performed by adding or removing FAGs from the cells, which is supposed to be performed less frequently due to the associated efforts for preparing the reconfiguration, adjusting the cell equipment (such as security fences) and recalibrating the robots in the new configuration. Regarding multiple cells, the systems can be reconfigured physically and logically by moving the FAGs from one cell to the other. As a result, the routing of the products can be modified and the utilization of the production cells can be adjusted. In this way, the described cell architecture realizes the key features of reconfigurable systems defined by Koren et al. [18]:

- **Modularity:** The system is composed of modular resources, e.g. the FAGs and the tooling.
- **Integrability:** The equipment of the FAGs and the 7-axis robots are complementary in function.
- **Customizability**
- **Customized flexibility:** The type of operations executed in a cell can be customized for the set of products to be produced.
- **Customized control:** By providing a generic control architecture, the automation of the cell can be adapted synchronous to the changing operations.

- **Convertibility:** Allows to change the cell with reasonable efforts (e.g. tooling).
- **Diagnosability:** The processing of products in batches allows to quickly detect quality deviations and avoid producing scrap by allowing to take corrective actions.
- **Mobility:** By removing the FAGs from one cell and integrating them into other cells, the location of process execution can be modified.

In this fashion, the modular structure makes it possible to reconfigure the cells in response to changing requirements of the system stakeholders. Yet, the size of the design space for the system architecture becomes difficult to manage when considering multiple cells for assembling multiple products with respectively several production steps and multiple instances of the FAG types, such as depicted in Figure 32. When allocating the two resources FAG₁ and FAG₄ to two different cells, the products requiring these resources (P₁ and P₃) have to be transferred between these production cells. Depending on their production volumes and number of batches, this decision could lead to a high number of transportation moves. Alternatively, when the FAGs are grouped based on the similarity of the processes of the products (e.g. P₁ and P₃ in one cell; P₂ in the other), the interdependencies could lead to other unwanted effects, such as low utilization of production cells, many transportation moves or high stocks. The figure illustrates the main degrees of freedom for designing the assembly system configurations: the number of cells, the number and allocation of products, and the number and allocation of FAGs. As the use-case company operates a high number of cells, the cells are coordinated mostly separately. This implies that when a product has to visit multiple cells for processing, the product is transported to the warehouse between the cell visits, since the local buffers next to the cells are not designed to stock more than the batch currently produced on the cell. As a result, each product produced on more than one cell causes additional transportation moves and stock levels. Hence, the two main characteristics of the problem the designers face are the number of opportunities for system design and the unique performance implications resulting from the interdependencies of the design elements.

Figure 32: System configuration problem

4.1.2 Scope of application

The scope of the DSET is that it should enable the users to explore various configurations of the reconfigurable assembly system. The approach should allow to consider the system configurations holistically from a structural perspective – encompassing all relevant production, logistics and storage processes – and also temporally, that is involving the current and the next phase of the system's lifecycle. For this purpose, the input for the DSET is the information the users can specify at the outset of the design process, also described as problem data in Figure 33, which includes the specifications of products, expectations for various developments of future demand, the specifications of the modular FAGs that can be clustered in the reconfigurable cells and the As-Is configuration of the system. Additional inputs are the basic description of the problem as perceived by the users and expressed by requirements for the design and performance of the solutions, as well as the knowledge base containing the procedures for automated synthesis, analysis and evaluation of the system configurations. While the knowledge base is static, the specific problem data, the requirements and the objectives are entered on runtime and can be iteratively refined. The output and result of the tool is embodied by the production system configurations selected by the users. Each of these configurations is composed of one or multiple production cells and characterized by its performance, as well as the allocation of equipment and products to the cells.

Figure 33: Input and output of the DSET

4.2 IMPLEMENTATION AND FEATURES

The DSET's main feature is that it can propose various sets of system configurations the users can iteratively explore and refine by including their requirements and objectives. Figure 34 visualizes the main steps and milestones of the concept. Initially, the users provide problem data, such as the specification of the products and production machines. If applicable, also the requirements for system configurations are entered at this moment. After that, an algorithmic procedure performs an initial problem analysis and uses one of three different design-rule sets to create a production system configuration. In the next steps, the performance of this solution is automatically analyzed and evaluated against the requirements. One result of this evaluation can be that the solution does not match the users' requirements. In this case, the system configuration is discarded and a novel configuration

synthesized instead (cf. (a), Figure 34). In the other case – if the solution matches the requirements – it is added to a solution repository that saves the system design as feasible part of the design space (b). The users initially request a specific number of solutions for their problem to be saved in the repository (e.g. 20, 50 or 100 solutions), which is referred to as a synthesis run. After the procedure has successfully generated the specified number of feasible system configurations, the solutions are visualized to the users. Subsequently, the users can assess and interpret the generated system configurations concerning their design and performance characteristics by using the GUIs of the DSET. In case they find solutions that exhibit unwanted properties, the users can refine their requirements and perform iterations (c). Otherwise, they can select one or multiple preferred solutions based on their strategic objectives (d).

Figure 34: Activities and milestones of the DSET-based design process

The technical design of the routine performed collaboratively by the users and the DSET reflects the concept for automating design synthesis suggested by Schotborgh et al. [30]. The DSET enables the users to iterate through the design process and compare the solutions proposed for different sets of requirements. This functionality is based on the interplay of four main building blocks:

1. The models of the problem components (i.e. products, equipment components, etc.).
2. The procedures that computationally synthesize the alternative designs based on the design-rule sets, analyze their performance and evaluate the solutions.
3. The production system model representing discrete system configurations.
4. The software application, i.e. the GUIs and functionalities that allow the users to specify their problem at the outset of the process, change the requirements for performing iterations and visualize the results for comparing and selecting solutions.

These components will be explained in the following four sections, followed by the description of their intended interplay when the decision-makers use the DSET.

4.2.1 *Design problem model*

The design problem is modelled and implemented based on the principles of object-oriented programming, using object classes as digital representations of the real-world objects and their properties. The model elements specified as input for the tool reflect the characteristics of products, the customer demand, the modular production resources and the As-Is configuration of the production system. The output of the tool are the synthesized production system configurations. The features defining these classes are summarized below, the detailed properties of the classes are described in Appendix A.6.

- **Products:** The products are the objects to be produced by the production system, represented by the technical specifications of their production processes and parameters describing their economic value, transport cost and optionally their product families.
- **Demand scenarios:** A demand scenario describes the expected demand volumes of the products. This information is used in combination with the products' technical specification to estimate the required capacity. Multiple demand scenarios can be formulated and used for the analysis of the synthesized system designs. Interesting scenarios are, for instance, lowest demand, average demand or highest demand.
- **Production resources:** The modular production resources and FAGs that are clustered to the production cells are characterized by their physical dimensions, processing speed and investment.
- **As-Is production system configuration:** The system configuration as existing on the shop-floor is described by the number of cells and the allocation of production resources and products to these cells. This information is used to estimate the effort for reconfiguring the system from the As-Is state to the configurations suggested by the DSET.
- **Synthesized production systems configurations:** The production system configurations represent the output of the tool and are the result of the synthesis procedures described in the following section. Their properties are explained in more detail in Section 4.2.4 and the related analysis methods presented in Appendix A.8.

4.2.2 *Algorithmic design procedures*

The design problem components are linked to each other by design rules, which express the dependencies between the various object types and their parameters. For instance, the production capacity of the synthesized system configurations is based on the capacity requirements of the products in the various scenarios. Multiple design rules are summarized to a design-rule set. The design-rule set conflates the information initially provided by the users several times to synthesize new system configurations. This section explains the three design-rule sets that the DSET can apply.

DESIGN-RULE SETS FOR CONFIGURATION TYPES The DSET uses three heuristic sets of design-rules to algorithmically generate a large variety of system configurations. Each design-rule set leads to one of the following configuration types:

- System configurations based on the product families (referred to as product-family configurations): In this configuration type, at least one production cell is synthesized for each product family specified in the product data, providing the production processes required by the respective family.
- System configurations composed of dedicated cells for multiple products (uni-cell configurations): In this configuration type the products are partitioned into different groups and a production cell with the required production processes is synthesized for each group.
- System configurations consisting of universal cells for multiple products (multi-cell configurations): For this type of configuration, first a number of cells is created and equipped with hardware. Then, the products are allocated to each cell.

The design-rule sets for generating these system configuration types imply varying constraints for allocating the products to the production cells. System configurations created by the product-family rule set require all products with the same product family to be produced in the same cell (left in Figure 35). The uni-cell rule set leads to system configurations where the products can be grouped in different ways and independent of their product families. However, they must be entirely produced in the same cell and in “one-stroke” production processes, in which no interruptions between the distinct production steps of the products are permitted (see center). In the configurations resulting from the multi-cell rule set, the production processes required for a product can be distributed across multiple cells. Consequently, products can be transferred between the production cells (see right, product 1), causing storage and logistics processes.

Figure 35: Example for input and results of the three design-rule sets

The rules in each design-rule set collectively resemble the three major phases of the design process, i.e. synthesis, analysis and evaluation. That means that each rule is associated to one of these phases. In the synthesis phase, the design of the system configurations is synthesized. Afterwards, the performances of the system configuration are calculated in the analysis phase. During the evaluation phase, the system configuration and its performances are compared to the requirements and, as a result, the solution is discarded or added to the design space. The three rule sets are explained in the following.

PRODUCT-FAMILY CONFIGURATIONS The synthesis of product-family system configurations starts with an empty solution object (cf. step 1, Table 3). The next step is determining the minimum number of cells required, which equals the number of product families specified as input information (2). On this basis, the maximum number of cells is determined by applying the cell scaling factor cSF (3). This scaling factor can be specified by the user to set an upper limit for the number of cells in the system. In the next step, the procedure randomly determines one value from the range between minimum and maximum and sets this value as the number of cells (4). The reasons for determining this and other design properties randomly are explained after the description of the configuration algorithms in this section. After determining the number of cells, the product families are allocated to the cells (5): in case the number of families and number of created production cells are equal, each product family and its products is allocated to one cell. If the number of cells is higher than the number of product families, each product family is allocated to one cell first and the remaining cells receive product families randomly afterwards. Based on this logical relation between products and production cells, the capacity requirement in the cells is determined for each year in the scenario that the user specified as normative for the design synthesis, referred to as the design scenario (6). To ensure the solutions are feasible, the maximum capacity requirement found in this scenario is used to determine the minimum number of resources required in the cell. Another user-specified factor – the FAG scaling factor fSF – is applied to determine the maximum number of each FAG type deployable in the cell (7). Afterwards, the value for the number of resources in the cell is randomly set between the minimum and maximum values (8). At this moment, the system design is fully specified and its performance can be analyzed in all demand scenarios (9). The results of the performance analysis are then evaluated by comparing them to the user requirements (10). In case the design or performance parameters are not in the required range, the configuration is discarded and a new one generated instead (11); otherwise it is added to the design space.

UNI-CELL SYSTEM CONFIGURATIONS Similar to the product-family configurations, the synthesis starts with generating an empty object (step 1, Table 4) and determining the minimum bound for the number of cells (2). Different to the product-family configurations, the minimum number of cells of the uni-cell configurations generally equals one. The maximum number of cells is determined by the number of products (3): since no transfers between cells are allowed in this type of configuration, the number of production cells must not be larger than the number of products. After randomly setting the number of cells (4), the products are randomly allocated to the cells (5): if the number of cells and products is equal, each cell receives one product; if the number of cells is smaller, each cell first receives one product and the remaining products are allocated to random cells subsequently. Then, the maximum capacity requirement is calculated. The principle is similar to the previously explained configuration procedure, yielding the minimum and

maximum values (6, 7) for the number of resources and the randomly instantiated value (8) for each cell. The performance analysis in the various scenarios (9), evaluation of the configuration (10) and result – retry or adding to the solution repository (11) – is performed in a fashion similar to the product-family configuration procedure.

MULTI-CELL SYSTEM CONFIGURATIONS The multi-cell configuration starts with the creation of the empty solution (step 1, Table 5) and by setting the minimum number of cells to one (2). Different to the previous rule sets, the maximum number of cells is determined by comparing the number of products and the number of distinct resource types and selecting the lower value (3). Based on this range, the number of cells is randomly set (4). As no products have been allocated so far, the maximum system-wide capacity requirement is calculated to determine the minimum (5) and maximum number of resources (6); then, the value is randomly determined respectively for each resource type (7). In the following step, these resources are randomly allocated to the cells (8): Each production cell first receives one random resource, before the remaining resources are randomly distributed. At this moment, the system's hardware configuration is specified, but the products are not yet allocated. Therefore the sequence for product allocation is determined by analyzing and sorting the products by the aggregate demand volume in the design scenario (9). The resulting sequence is then used for processing the products in descending order (10): for each product, the specific process sequence is compared to the system hardware configuration and all logically feasible routings are listed (11). These combinations are then sorted by the number of transports they induce for the considered product, aiming to select a combination that causes low stock-levels, transports and associated cost (12). Starting with the routing that involves the least cells, the capacity-requirement of the product is determined (13). If sufficient capacity is available in all involved cells, the routing is feasible and the product is allocated to these cell(s); in case it fails, the routing is discarded; in case all routings have failed, the configuration is discarded and another attempt is made. Once all products were successfully allocated, the performance of the configuration is analyzed in the demand scenarios (14) and evaluated similarly to the other configuration types (15), resulting in a retry or an inclusion of the new configuration in the solution space (16).

Table 3: Steps and design rules of the product-family configuration type

Step #	Step	Product-family rules	Phase
1	Start	Create new (empty) solution object	Synthesis
2	Determine minimum number of cells min_C	$min_C = \text{numberofproductfamilies}$	
3	Determine maximum number of cells max_C	$max_C = cSF * min_C$	
4	Set number of cells	Set random value for num_C in $[min_C, max_C]$	
5	Product allocation	If $num_C = min_C$: assign each family and related products to one cell; Else: assign each family to one cell and randomly assign product family/families and related products to remaining cells	
6	<i>For each cell</i> : Determine minimum number of production resources for each process (design scenario)	<i>For each cell</i> : $cap_{max_R} = \text{maximum capacity requirement of all resources in scenario}$; Minimum number of resources required for process R: $min_R = \text{fraccap}_{Max_R} cap_{year}$ (rounded up)	Analysis
7	<i>For each cell</i> : Determine maximum number of production resources for each process	<i>For each cell</i> : Maximum number of resources required for process R: $max_R = min_R * (tSF + 1)$ (rounded up)	
8	<i>For each cell</i> : Set number of resources	<i>For each cell</i> : Set random value for num_R in $[min_R, max_R]$	
9	Performance analysis	Calculate all performances of the system configuration in each scenario	Evaluation
10	Design & performance evaluation	Compare design parameters and performances with user-specification	
11	End	If evaluation = ok: add system configuration to solution space Else: discard system configuration and retry (go to step 1)	

Table 4: Steps and design rules of the uni-cell configuration type

Step #	Step	Uni-cell rules	Phase
1	Start	Create new (empty) solution object	Synthesis
2	Determine minimum number of cells $\min_C$	$\min_C = 1$	
3	Determine maximum number of cells $\max_C$	$\max_C = \text{number of products}$	
4	Set number of cells	Set random value for num_C in $[\min_C, \max_C]$	
5	Product allocation	If $\text{num}_C = \text{number of products}$: assign each product to one cell; Else: assign one product to each cell, then randomly assign remaining product(s)	
6	For each cell: Determine minimum number of production resources for each process (design scenario)	For each cell: $\text{cap}_{\max_R} = \text{maximum capacity requirement for resource R in scenario}$ Minimum number of resources required for process R: $\min_R = \text{fraccap}_{\max} \text{cap}_{\text{year}}$ (result rounded up)	
7	For each cell: Determine maximum number of production resources for each process	For each cell: Maximum number of resources required for process R: $\max_R = \min_R \cdot \text{fSF}$ (rounded up)	
8	For each cell: Set number of resources	For each cell: Set random value for num_R in $[\min_R, \max_R]$	
9	Performance analysis	Calculate all performances of the system configuration in each scenario	Analysis
10	Design & performance evaluation	Compare design parameters and performances with user-specification	Evaluation
11	End	If evaluation = ok: add system configuration to solution space Else: discard system configuration and retry (go to step 1)	

Table 5: Steps and design rules of the multi-cell configuration type

Step #	Step	Multi-cell rules	Phase
1	Start	Create new (empty) solution object	Synthesis
2	Determine minimum number of cells min_C	$min_C = 1$	
3	Determine maximum number of cells max_C	If number of products > number of resource types $\Rightarrow max_C = \text{number of resource types}$; Else: $max_C = \text{number of products}$	
4	Set number of cells	Set random value for num_C in $[min_C, max_C]$	
5	Determine minimum number of production resources for each process (design scenario)	$cap_{maxR} = \text{maximum capacity requirement for resource R scenario}$; Minimum number of resources required for process R: $min_R = frac{cap_{maxR}}{cap_{year}}$ (result rounded up)	
6	Determine maximum number of production resources for each process	Maximum number of resources required for process R: $max_R = min_R \cdot f_{SF}$ (rounded up)	
7	Set number of resources	Set random value for num_R in $[min_R, max_R]$	
8	Hardware allocation	Allocate one random resource to each cell; Allocate remaining resources randomly;	
9	Sort products	Sort products by aggregate demand volume in the design scenario	
10	Loop Step 1: Pick product	Pick product with highest demand	
11	Step 2: Analyze allocation opportunities	Get all routing options for the product	
12	Step 3: Sort allocation opportunities	Sort opportunities by number of inter-cell transports;	
13	Step 4: Try allocation End loop	Begin with routing option with least number of inter-cell transports: test capacity-related feasibility of allocation: If successful: allocate current product and pick next product (go to step 10); If failing: discard routing option and try next (go to step 13); If all options infeasible : discard system configuration and retry (go to step 1);	
14	Performance analysis	Calculate all performances of the system configuration in each scenario	Analysis
15	Design & performance evaluation	Compare design parameters and performances with user-specification	Evaluation
11	End	If evaluation = ok: add system configuration to solution space Else: discard system configuration and retry (go to step 1)	

SAMPLING STRATEGY REQUIREMENTS An important aspect with regard to the presented synthesis procedures is that they are designed to be capable of generating a large number of different system configurations. In fact, the number of possible solutions is dependent on the problem specification by the users and hence can encompass billions of solutions. To enable the users to visually compare the suggested solutions and maintain the expressiveness of the visualization, the number of presented system configurations has to be limited to a level the users perceive as suitable. The users can therefore perform an arbitrary number of synthesis runs and decide whether 20, 50 or 100 solutions should be generated in each run. This pre-selection of few out of the many possible solutions represents a sampling strategy. The requirements for the sampling strategy are the same as for the DSET, i.e. the sampling strategy must be applicable without long waiting times and provide solutions matching various requirements and objectives of the users.

RANDOM DESIGN SYNTHESIS Based on these specifications for the sampling procedure, it was decided to randomly instantiate specific design parameters (number of cells; number of FAGs; allocation of product families, products and production steps) as it implies a non-directed design procedure. Since the users can provide bounds for some of the properties that are randomly determined by specifying requirements (number of cells, number of FAGs) and also other randomized aspects depend on the user input (e.g. number of products), the term of semi-random design synthesis is used in the following. The semi-random instantiation of design properties represents an alternative to exhaustive search procedures that can imply long waiting times when generating and assessing a large number of possible solutions. The ramifications of this approach are that in case the users specify narrow requirements (e.g. utilization between 90% and 95%), many randomly suggested solutions might not fulfill the requirement and are therefore discarded. As the DSET's routines can only visualize the generated solutions once all solutions of the synthesis run were synthesized, the consequence of semi-random synthesis and narrow requirements could be long waiting times until the synthesis runs are completed. To prevent those waiting times, the maximum number of attempts to synthesize solutions is limited to a maximum of four times the requested number of solutions. As a result, it amounts to a maximum of 80 solution generation attempts for a synthesis run of 20 solutions, 200 attempts for 50 solutions and 400 for 100 solutions. Once the maximum is reached the solutions generated until that moment will be displayed, which can result in less solutions than initially requested.

SCALING FACTORS In this context, the scaling factors fulfill an important function. The semi-random synthesis implies that the generation of system configurations is not directed, i.e. the procedures are not designed to purposefully minimize the investment or maximize capacity utilization. Hence, the procedure can generate system configurations with various characteristics that satisfy the preferences of the users, such as specific ranges of aspired capacity utilization/reserve. These preferences can be formulated by integrating requirements that are used by the algorithms to evaluate the synthesized system configurations after their analysis. Since the number of attempts for synthesizing system configurations is limited it can happen that only very few solutions with the aspired characteristics result from a run (see Figure 36, left). Therefore the scaling factors can be used to control the two core design parameters that determine the capacity and indirectly affect most other design and performance parameters in the system. The cell scaling factors cSF allows to control the maximum redundancy of cells in the product-family rule set, whereas the FAG scaling factor fSF determines the maximum redundancy of FAGs in all rule sets. By default,

both factors are set to $cSF = fSF = 10$, implying that the number of cells for the product family configurations and the number of machines for all rule sets can amount to 10 times the minimum requirement determined in the initial problem analysis, enabling a large variety of the resulting system configurations. In case no or only very few solutions result from a synthesis run, the users can specify requirements for the synthesis run and experimentally test various combinations to increase the chance of obtaining solutions matching their requirements. Hence, the semi-random design synthesis aims at an efficient design automation process, whereas the scaling factors represent an opportunity to increase the effectiveness by customizing the design synthesis procedure.

Figure 36: Solutions generated with default scaling factors (left) and constrained factors (right)

4.2.3 Interfaces and software specifics

The DSET's graphical user interfaces (GUIs) are designed to allow the users to calibrate the described algorithmic procedures and understand their results. On the input side, the requirements and the scaling factors are specified. On the output side, the synthesized solutions and their performances are presented.

GUIs FOR INITIALIZING AND CALIBRATING THE SYNTHESIS The GUI on the input side consists of three main panes (see Figure 37): The project pane shows the individual synthesis runs performed and allows to adjust the requirements for the next run (element 1). The pane in the center is the place where the users can integrate their specific requirements (2). In the illustrated example, the synthesis rule set to be applied is specified by adding an expression for the property called 'Meta', leading to product-family-based system designs. Other possible form of specifying the requirements are individual values or value intervals for numerical properties (e.g. property 'Investment'). To add further requirements, the applicable properties can be selected from the pane on the right (3) that lists all model properties present in the knowledge base. The required value for the property can be specified in the central pane (e.g. property 'Investment' should lie in the range between EUR 0 and EUR 750.000). Once all properties have been added and respective values assigned, a synthesis run can be performed with these requirements by choosing the requested number of system configurations in a drop down menu (4).

GUIs FOR EXPLORING AND SELECTING SOLUTIONS On the output side, the central pane of the GUI provides the functionalities for exploring the design space (see Figure 38): one opportunity to visualize the system configurations generated through the algorithmic procedure is to display them in x-y scatterplots (element 1). In such a chart, each system configuration (2) is represented by a dot that is located at the values of the specific

property chosen to be mapped on the x- and y-axis (3). The properties to be visualized on the axes can be customized. By choosing another property combination, the relative position of the solutions changes according to their values for these properties, allowing to assess the position of the system configurations in relation to each other while assessing various design or performance properties. Another way to compare the solutions is the table-view (4), in which each row describes the property values of one system configuration (5). By clicking on the column header containing the property of interest, the solutions are sorted based on the property value, allowing to identify the solutions with the minimum and maximum property values. The dropdown menu in (6) allows to choose another demand scenario, which updates the x-y-plot and the table by displaying the performance of the system configuration in the chosen scenario. Also, the color of solutions can be used to indicate parameter values (7). The result is that the symbols representing the system configurations in the x-y-plot are colored according to their property values and can be interpreted based on the color legend (8) that dynamically adapts to the values of the solutions under investigation. Also the symbols used to represent solutions can be customized (9, 10) to distinguish the solutions generated in different synthesis runs or the solutions that are Pareto-efficient for specific properties. The latter can be particularly helpful if more than three properties should be evaluated concurrently and is an often applied selection approach in multi-objective problems [113]. The inherent calculations for this feature are explained in Appendix A.7. The users can mark the properties relevant for the Pareto-quality in a list showing the system properties (11). For each solution property, the users can indicate if the minimum or the maximum value describes the preferred solution characteristic. The solutions with Pareto efficiency are visualized by crosses, in contrast to the dots that represent the non-efficient solutions.

CUSTOMIZABILITY OF THE CHARTS AND COMPARISONS The main idea of the exploration GUI is to provide functionalities that enable the users to customize their perspective of the design space and its constituent solutions. The users can select the solution sets generated in multiple runs to be displayed in the same graph (12) and then assess and compare the system configurations with regard to the properties they find particularly relevant. Multiple scatterplots and tables can be visualized in parallel for comparing multiple properties at once (see Figure 39). When the users mark one or multiple solutions in the scatterplot, the respective entries in the table and other scatterplots are highlighted. If another property type should be visualized, the solutions stay highlighted so that they can be traced when comparing them regarding various design and performance properties. Similarly, choosing a different scenario updates the solutions' relative positions in the graphs, but the selected solutions remain identifiable. Hence, the users can identify the most suitable solutions for their objectives based on the visual contrast in the x-y-plots, the implicit ranking functionality of the table-view, the color of solutions (value) and their shape (Pareto-efficiency). While the Pareto-efficiency is a quantitative approach to evaluating the solutions, the functionalities for visual comparison are suited for intuitive evaluation.

Figure 37: Interface for initializing synthesis runs and specifying requirements

Figure 38: Customizable interface for exploring, comparing and selecting system configurations

Figure 39: Customizable graphs for evaluating the generated solutions

SOFTWARE SPECIFICS The previously presented production system model and procedures were developed by the author and implemented in C# programming language, while the software application that visualizes the results in the GUIs and manages the interactions with the users – i.e. for entering requirements or customizing the graphs – was implemented as a Windows Forms application at the Department of Design, Production and Management by Ir. Hans Tragter and Clemens Wolters. The implemented DSET architecture separates the design problem-related software components from the software components for visualization and interaction. The motivation for this modularization of the software is to make the related components functionally independent, allowing for efficient development and maintenance of both [97, 98, 114]. The three components were developed in close collaboration to achieve the described functionality.

4.2.4 *System configuration properties and requirements*

The result of the synthesis procedures presented in Section 4.2.2 are the various production system configurations. Each synthesized system configuration has properties that describe it from the perspectives of various stakeholders (see Table 6). The decision-makers can use the design, technical, economic and synthesis-related properties to compare the system configurations in the GUIs. For instance, they can compare the solutions with regard to the number of cells and occupancy of the shop-floor area, the systems' utilization rates, investments and cost of reconfiguring their system to the prospective solutions. The detailed description of the meaning and the analysis rules for calculating the numerical performance of all properties are presented in Appendix A.8.

The formulation of requirements for these properties plays an important role in generating system configurations that reflect the decision-makers' intentions. The properties with asterisk in Table 6 can be prescribed by the users and are an input for the DSET that influence the synthesis procedure. By entering a requirement for the property 'Meta', the design-rule set to be used by the algorithmic synthesis procedure is specified. Figure 37 shows an example where the product-family rule set is used. Also the values of the cell and fag scaling factor are an input for the procedure that have direct influence on the design of the system configurations. Hence, formulating requirements for these factors makes it possible to customize the values and create system configurations with non-default factors.

In contrast to that the properties without asterisk are results of the synthesis and analysis procedures. Hence they are relevant only once the design of each system configuration and the performance analysis are finished. The evaluation procedure then simply compares the actual property value to the value (range) of the requirement (e.g. actual value: $x = 2$, requirement $rx: 1 < x < 3$; evaluation: x fulfills rx ?; result: *true*). This evaluation is performed for all properties for which requirements are specified. The solution is accepted if all requirements are fulfilled or discarded otherwise. This approach allows to easily evaluate numerical values by entering specific values or intervals, yet the formulation of requirements for properties formatted as strings is not possible (only exception is the property 'Meta' since the possible values are predefined). As a result, requirements can be specified for all configuration properties without superscript '1' in Table 6, e.g. for the number of cells, the investment or the utilization of the resulting solutions. This makes it possible to indirectly influence the design properties by using requirements to constrain numerical properties whose values are due to characteristic configuration designs. For instance, a requirement can be formulated for the intercell-transport indicator to make system configurations with a widely distributed allocation the products infeasible. Therefore, many

Table 6: Properties of synthesized production system configurations

Property		Property type	Property		Property type
<i>Design</i>	Meta ⁱ	String	<i>Economic</i>	Equipment value	Double
	Cell scaling factor ⁱ	Double		Investment	Double
	FAG scaling factor ⁱ	Double		Reconfiguration cost	Double
	System hardware design ¹	String		Storage cost	Double
	System configuration ¹	String		Logistic cost	Double
	Routing ¹	String		Direct production cost	Double
	Number of cells	Integer		Changeover cost	Double
	Number of FAGs	Integer		Reprogramming cost	Double
	Area	Integer		Total cost	Double
<i>Technical</i>	Stock-level	Integer	<i>Syn.*</i>	Non-utilized value	Double
	Transports	Integer		Run ID ¹	String
	Intercell transport indicator	Double		PS name ¹	String
	System utilization	Double			
	Cell-specific utilization rates ¹	String			
	Year-related cell utilization rates ¹	String			

ⁱ Property is input for the synthesis procedure

¹ Property value difficult to constrain by requirements

* Syn. = synthesis-related properties

properties that can be used to compare system configurations can be also used to formulate requirements. For the few properties that cannot be directly specified, other related properties can be used to indirectly address them.

4.2.5 Usage of the DSET

The intended usage of the DSET is as follows: in the first step, the users provide the data specifying their problem, described by the products, production machines, As-Is system configuration and the demand scenarios (see step 1, Figure 40). Then, the DSET is initialized by loading this data, as well as the knowledge base that contains the model described in Section 4.2.1 and the algorithmic procedures introduced in Section 4.2.2 (step 2). Then the users have the opportunity to specify requirements (step 3). Based on this problem formulation, the users execute the first synthesis run, in which the procedures attempt to produce 20, 50 or 100 system configurations for the specified problem and display the design candidates and their performances in the GUIs (step 4). Using the functionality described in Section 4.2.3, the users explore the generated system configurations by comparing and evaluating them (step 5). The users can then perform multiple runs with the same requirements to obtain further system configurations. Alternatively, they can perform further synthesis runs with changed requirements. The automated synthesis and analysis procedures then propose new configurations and automatically evaluate every solution based on the entered requirements. Hence, by iterating through the steps 3 to 5, the solutions resulting from various combinations of requirement can be assessed. After identi-

fying and selecting the most suitable solutions (step 6), the users can proceed with the next step of implementing one of the previously selected system configurations. Alternatively, they can perform further iterations and test the implications of related problems (step 7), e.g. expanding or reducing the set of products, assigning the product families differently or considering different assumptions for the performance of the FAGs.

Figure 40: Intended process flow of the DSET

The logic of the support tool with regard to the design space is illustrated in Figure 41. The problem data provided by the users at the outset is associated to a very large space of potential system design alternatives. Out of this large number of potential solutions, the algorithmic synthesis procedure makes accessible the theoretical configuration space (abbr. TCS). From this pool of potential solutions, the synthesis procedures create discrete instances semi-randomly and compare them to the users' requirements. The result of this evaluation are the solutions matching the users' requirements, which are visualized in the DSET. Afterwards, the users can either iterate and obtain additional solutions by changing

Figure 41: Rationale of the DSET

the solution requirements and problem specifics or integrate their objectives to choose the final solution(s).

4.3 ENVISAGED FULFILLMENT OF THE SUPPORT TOOL OBJECTIVES

The DSET's features were designed to provide for reaching the objectives presented in Section 3.1.3. Figure 42 illustrates the intended interplay of the features and shows how they are manifested in the DSET, i.e. whether they are based on the model/procedure or the GUI. The following paragraphs outline the intended interaction between the features that should eventually lead to the effectiveness and efficiency of the design approach.

Figure 42: Rationale of the DSET

VARIETY OF CONFIGURATIONS The variety of configurations should be increased in comparison to manual design processes by applying three design-rule sets capable of generating a broad range and different types of system configurations.

NUMBER OF CONFIGURATIONS An increase of the considered number of system configurations should be achieved by automatically synthesizing the configurations based on the design-rule sets. To make sure that the solutions represent relevant concepts from the perspective of the users, the requirement-based filtering and iteration supports this objectives by filtering out the irrelevant configurations.

SPEED OF APPLICATION All features of the design synthesis model and the GUI-related are supposed to make the DSET a fast approach to synthesis, analysis, evaluation and selection of system designs.

APPLICABILITY OF CONFIGURATIONS The opportunity to specify requirements for the configurations should ensure that the proposed solutions are relevant concepts for the users' problem and can be implemented in practice.

QUALITY OF CONFIGURATIONS The quality of the system configurations should be provided by enabling the users to select their preferred configurations from the synthesized sets, in which all solutions fulfill their requirements. The customizable plots and tables should support them in comparing and selecting solutions. Moreover, the configurations' sensitivity to the demand scenarios can be assessed and superior solutions can be identified visually in the graphs and tables, additionally supported by the feature that highlights the Pareto-efficient solutions.

CUSTOMIZABILITY OF THE SUPPORT TOOL By providing customizable requirements and objectives in combination with algorithmic procedures that are capable of suggesting solutions for user-defined problem information, the users should be able to apply the DSET to consider the solutions of various problem formulations, i.e. various combinations of problem data, requirements and objectives.

EFFICIENCY OF THE DESIGN PROCESS The measure for the efficiency of the design process is the required efforts for evaluating a number of candidate solutions in a certain timespan. The suggested features should enable the users to evaluate a large number of distinct and applicable configurations in a short time and with low efforts with regard to the entire sequence of activities from preparation to the selection of solutions.

EFFECTIVENESS OF THE DESIGN PROCESS The effectiveness of the design process is described by the properties of the selected solution(s). The DSET should enable the users to consider many, various and relevant solutions for their related problem statements. Once the quality and implications of the suggested configurations has been evaluated by the users, the most suitable proposal(s) can be selected.

4.4 DISCUSSION

CONCATENATING PROBABILITIES An important implication of the suggested synthesis approach is that the probabilities of instantiating the various solutions are not necessarily the same. In all configuration types, the values for the design properties – the number of cells, the allocation of products and resources – are set after one another. This leads to concatenating probabilities for instantiating the design properties, as shown in Figure 43: if the number of cells is set to 1, all products will be allocated to the single cell since there is only one option; if the number of cells is four, there can be more opportunities for allocating the products (10 in the example). Presupposing that the semi-random assignment of properties makes each possible property value equally likely, the only one-cell configuration has a total occurrence probability of 25%, whereas the distinct four-cell configurations have average probabilities of 2,5% respectively. The specific probability of each configuration is referred to as the synthesis probability in the following. Its quantitative implications, the link to the scaling factors and the semi-random synthesis are shown in Chapter 5.

USE CASE ORIENTATION Furthermore, it should be noted that the described model and algorithmic design procedures are specifically designed for the situation found in the use-

Figure 43: Synthesis probabilities of individual solutions

case company of the Robust PlaNet project. Hence, the problem definition and the scope of the application were largely defined by the company. Similarly, the properties of the system configurations and the availability and level of detail of information describing the initial problem components (products, scenarios, machines) are based on the company's demands. The related research and knowledge elicitation was performed by mechanical engineering students who completed their master graduation assignments in the company with focus on generating a basis for the development of the DSET [115, 116].

4.5 CONCLUSIONS

This chapter describes the design and features of the DSET, a computational tool to support the users in exploring the configuration space of the reconfigurable cellular assembly system. The DSET symbolizes the answer to the second research question (*How can the design of reconfigurable production systems be supported by means of automated design synthesis?*). The implemented model reflects the industrial system design problem considered in the use-case of the Robust PlaNet project. The main functionality of the DSET are algorithmic procedures that can automatically suggest and analyze various production system configurations, in which the production resources and products are allocated in various ways to one or multiple production cells. An automated evaluation procedure filters the solutions that do not match the users' requirements. The DSET's GUIs enable the users to iteratively assess the generated solutions and change the requirements, allowing to increase the range of considered solutions. Furthermore, features that support the users in identifying particularly suitable solutions in the generated sets allow them to converge towards solutions that fulfill their objectives. In this fashion, the DSET represents an approach to fulfilling the requirements for support tools elaborated in the previous chapter.

Part IV

DESCRIPTIVE STUDY II

The following chapters describe the evaluation of the DSET conducted to indicate its characteristics with regard to the objectives of the framework as well as the applicability and usefulness of the approach. Evaluations of support tools performed as part of engineering design research are characterized by three different stages [45]:

- Support evaluation: The objective of this stage is to determine whether the support tool works properly and as intended from a functional perspective. A working prototype that matches the description in Chapter 3 is available, which implies that this step was successfully completed.
- Application evaluation: The objective of the application evaluation is to determine whether the proposed DSET can be used for the task it was designed for and addresses the key factors. In the presented research, these factors are the direct objectives of the DSET's features (cf. Figure 42).
- Success evaluation. In the success evaluation, it is determined whether the support tool allows to reach the intended measurable result. The aspired results of this research are the effectiveness and efficiency of the design process.

The following two chapters discuss the achievement of the DSET's objectives from two perspectives. First, Chapter 5 assesses the DSET's characteristics by taking a detailed look at the synthesized system configurations and describing the behavior of the DSET with regard to various problem statements, answering RQ3. Secondly, Chapter 6 describes the evaluation of the DSET's effects by practitioners of the use-case company, focusing on their perception of its applicability and usefulness and answering RQ4. In combination, these two chapters present the characteristics of the design support approach and make it possible to evaluate its applicability, usefulness and success. To valorize the presented research, Chapter 7 discusses opportunities to combine the DSET with state-of-the-art support approaches and points out how the approach could be enhanced. Following this, Chapter 8 discusses the research and concludes the thesis.

EVALUATION OF THE DSET'S CHARACTERISTICS

This chapter evaluates the DSET by assessing the solutions obtained for specific problem statements. The results show the implications of the approach and provide answers for RQ3: *What are the characteristics of the suggested approach to design automation?*.

5.1 RESEARCH APPROACH

The DSET was examined with respect to the intended effects on the resulting variety and number of system configurations, as well as its speed and customizability. Three aspects are particularly relevant with respect to these objectives. Since the DSET automatically proposes various system configurations with the objective to enable an exploration and selection of alternative designs, the decision-makers should know to which extent they can expect the tool to present all potential design options, that is, the design alternatives that are feasible and fulfill their requirements. The related evaluation question (EQ1) is formulated as: *Which variety of system configurations is synthesized by the procedures in comparison to the potential options?*

Furthermore, the DSET enables the users to iteratively define system requirements, that is define requirements, generate solutions, redefine requirements, generate solutions, and so forth. In this regard, the effects of changing problem information and requirements on the generated solutions has to be known (EQ2): *How do changes of the user input affect the variety of system configurations?*

Lastly, also the DSET's capability to propose feasible alternatives in various design situations must be assessed, answering the question (EQ3): *Which variety of system configurations can be obtained when facing various design situations?*

Each of the issues will be discussed in their respective sections, EQ1 in Section 5.2, EQ2 in Section 5.3 and EQ3 in Section 5.4. The DSET enables the users to eventually select the most suitable system configurations from the automatically generated sets, which means that the aspired result of the DSET's algorithm is not a single system configuration, but multiple feasible design solutions that form the computational configuration space (abbr. CCS; visualized in Figure 41). This peculiarity is reflected in the design of experiments, which does not focus on individual solutions but discusses the characteristics of the sets of system configurations generated by the DSET. To facilitate an overview and comparison of the experiments and their results, the main characteristics of each experiment and the obtained solutions are summarized in tabular form in the respective sections.

please note: The graphs presented in Sections 5.2 and 5.3 show characteristics of the solutions to support the readers' understanding of the evaluation results. Those graphs are not visualized by the DSET. In contrast to that, the graphs in Section 5.4 resemble chart types that are visualized in

the DSET. Since screenshots of the software have a limited resolution, these graphs were reproduced for high-quality presentation.

5.2 COVERAGE OF THEORETICAL SOLUTIONS BY THE SYNTHESIS PROCEDURES

Firstly, the exhaustiveness of the actual suggested variety and number of solutions is compared to the design solutions that can be theoretically generated by the model. The tool can be considered as a window that allows the users to peek into the design spaces shaped by their requirements. As highlighted in Section 4.4, the design procedures randomize particular design aspects, which can affect the synthesis probability of the solutions and thus the variety of the synthesized solutions. Hence, the effects of this randomization on the resulting quantitative and qualitative characteristics of system configurations are examined.

Five experiments were designed to evaluate the number and variety of the solutions and their characteristics. Experiments 1 and 2 in Section 5.2.1 quantify the variety of the instantiated values for two design properties, the number of cells and the allocation of equipment. The examination of these two design variables should illustrate the effects of randomization representatively for all randomly determined properties. Experiments 3 to 5 described in Section 5.2.2 examine the total variety of the resulting designs of the system configurations and corresponding performances to assess the effects of concatenated property randomization. After that, Section 5.2.3 discusses the findings.

To provide a reference for evaluating the design solutions that were synthesized in the experiments and which form the computational configuration space, all system configurations possible for the degrees of freedom of each experiment were manually enumerated and give shape to the theoretical configuration space (TCS). In each experiment, the solutions of the CCS are analyzed and compared to the ones of the TCS (see Figure 41), allowing to determine the coverage of the synthesis procedures. Eventually this enables to assess the variety of the synthesized solutions and quantify the effects of the randomization.

5.2.1 Variety of solution characteristics

This section examines the variety of the values of two solution properties. Experiment 1 focuses on the number of cells, whereas experiment 2 describes the allocation of FAGs.

Experiment 1: Number of cells

To characterize the TCS and the CCS, the means and standard deviations of the number of cells are determined. The number of cells of the TCS is described by the theoretical properties, the expected value $E(x)$ and standard deviation σ_E . The number of cells of the CCS is calculated based on the values obtained in the synthesized set, described by the arithmetic mean $\bar{x}$ and the actual standard deviation σ_A . Table 7 gives an overview of the experimental set-up and the observed results. Additionally, Appendix A.9 illustrates the input data used for the experiment.

Due to the product-family rule set and the applied scaling factors, three values x_i are possible for the number of cells – 2, 3 and 4 –, resulting in a probability of $p(x_i) = 1/3$ for each outcome. Thus, the expected value $E(x)$ for the number of cells is:

$$E(x) = \sum x_n \cdot p(x_n) = 2 \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} + 4 \cdot \frac{1}{3} = 3 \quad (5.1)$$

Table 7: Profile of experiment 1

<i>Objective of experiment</i>	Variety of the number of cells
<i>Design rules</i>	Product-family rule set
<i>Products to be produced</i>	Four products with two to four production steps; Product families: PF1 → P1, P2; PF2 → P3, P4
<i>Distinct production technologies</i>	Four different process types
<i>Scaling factors</i>	cSF=2; fSF=1
<i>Number of distinct solutions possible (TCS)</i>	3 system configurations
<i>Number of solutions synthesized</i>	50 system configurations
<i>Number of distinct solutions synthesized (CCS)</i>	3 system configurations

Subsequently, the CCS-set was analyzed for the arithmetic mean of the number of cells in the system configurations. The set contains 19 configurations with two cells, 11 configurations with three cells and 20 configurations with four cells, resulting in an arithmetic mean $\bar{x}$ of:

$$\bar{x} = \frac{1}{n} \cdot \sum_{i=1}^n (n_{x_i} \cdot x_i) = \frac{19 \cdot 2 + 11 \cdot 3 + 20 \cdot 4}{50} \approx 3,02 \quad (5.2)$$

The expected standard deviation σ_E for the number of cells is calculated as follows:

$$\sigma_E = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (x_i - E(x))^2} = \sqrt{\frac{1}{2} \cdot \sum_{i=2}^4 (x_i - 3)^2} = 1 \quad (5.3)$$

The actual standard deviation σ_A of the CCS was:

$$\sigma_A = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (x_i - \bar{x})^2} = \sqrt{\frac{1}{49} \cdot [19 \cdot (2-3)^2 + 11 \cdot (3-3)^2 + 20 \cdot (4-3)^2]} \approx 0,89 \quad (5.4)$$

Since $\bar{x} \sim E(x)$, the number of cells of the synthesized solutions matches the expected value. The expected and the actual standard deviation show comparable, yet not so close values, with $\sigma_E = 1$ and $\sigma_A \approx 0,89$. Regarding the difference of approximately 0,11 between these values it should be noted that the probability $p(x_n)$ and the expected value $E(x)$ are statistics that should result as the average of a large number of observations. Observing $\bar{x} = E(x)$ and $\sigma_E = \sigma_A$ based on a sample of only 50 synthesized solutions is not very probable.

Experiment 2: Variety of FAG allocation

The other solution property evaluated for representative purposes is the equipment allocation. Table 8 shows the basic parameters for the experiment. As the numbers of required equipment types differ between the product families, the possible range is $n_{C1} = 16$ different cell configurations for each cell of product family 1 and $n_{C2} = 8$ distinct cell configurations for product family 2 (cf. Appendix A.10), resulting in a total of $n_{Total} = 8 \cdot 16 = 128$ potential configurations for the indicated requirements.

The 50 generated system configurations resulted in 43 distinct ways of allocating FAGs. The resulting frequency of the configurations in the CCS can be seen in Figure 44 (top). The

Figure 44: Observation frequency of the distinct system configurations of the product-family rule set in the TCS (x-axis) and CCS (bars) in E2 (top graph) and E3 (bottom graph)

Table 8: Profile of experiment 2

<i>Objective of experiment</i>	Variety of FAG allocation
<i>Design rules</i>	Product-family rule set
<i>Products to be produced</i>	Four products with two to four production steps; Product families: PF ₁ → P ₁ , P ₂ ; PF ₂ → P ₃ , P ₄
<i>Distinct production technologies</i>	Four different process types
<i>Scaling factors</i>	cSF=1; fSF=2
<i>Number of distinct solutions possible (TCS)</i>	128 system configurations
<i>Number of solutions synthesized</i>	50 system configurations
<i>Number of distinct solutions synthesized (CCS)</i>	43 system configurations

comparison of the TCS and the CCS shows that some system configurations are instantiated multiple times (e.g. IDs 6 and 30), others only once (e.g. ID 7 to 24), and a wide number of configurations is not instantiated at all (represented by IDs 44 to 128).

5.2.2 Variety of solutions

This section focuses on the variety of the obtained system configurations caused by the concatenated randomization of design aspects. Each design-rule set is evaluated separately for the number and variety of solutions found. Experiment 3 evaluates the product-family rule set, Experiment 4 the uni-cell rule set and Experiment 5 the multi-cell rule set.

Experiment 3: Solution variety of the product-family rule set

The experimental data for evaluating the solution variety of the product-family rule set is displayed in Table 9.

The TCS resulting from the scaling factors consists of 128 distinct system configurations. The CCS is analyzed with respect to the variety of embodiment and performance parameters. The 250 synthesized solutions resulted in 109 distinct system configurations. Figure 44 (bottom) shows the absolute instantiation frequency of the distinct configurations. Some candidates are instantiated multiple times in the solution set, others occur once and some

Table 9: Profile of experiment 3

<i>Objective of experiment</i>	Variety of the product-family configurations (FAG allocation; performances)
<i>Design rules</i>	Product-family rule set
<i>Products to be produced</i>	Four products with two to four production steps; Product families: PF ₁ → P ₁ , P ₂ ; PF ₂ → P ₃ , P ₄
<i>Distinct production technologies</i>	Four different process types
<i>Scaling factors</i>	cSF=1; fSF=2
<i>Number of distinct solutions possible (TCS)</i>	128 system configurations
<i>Number of solutions synthesized</i>	250 system configurations (5 sets with 50 solutions each)
<i>Number of distinct solutions synthesized (CCS)</i>	109 system configurations

system configurations cannot be found. Moreover, it can be seen that the randomization makes the runs differ from one another in terms of the generated configurations.

Moving the focus towards the performance of system configurations in the CCS, Figure 45 shows the maximum (left) and the minimum values (right) of the system properties investment and utilization that can be found among the system configurations of the individual runs. The statistics suggest that using the product-family rule set to synthesize multiple solution sets results in system configurations with comparable features regarding the minimum and maximum values for utilization and investment, yet identical extremes can be only observed for the maximum values of run 1 and 3 and the minimum values of runs 2, 3 and 4. It can be therefore stated that in the presented example, the semi-random assignment of the design aspects made each set a unique record of the possible system configurations.

Figure 45: Maximum (left) and minimum property values (right) observed in the synthesis runs (product-family rule set)

Experiment 4: Solution variety of the uni-cell rule set

The configuration problem for evaluating the uni-cell rule set is presented in Table 10. As the input dataset describes five products, the uni-cell rule set can generate configurations with one cell (all products produced in the same cell) up to five cells (each product

Table 10: Profile of experiment 4

Objective of experiment	Variety of the uni-cell configurations (product and FAG allocation; performances)
Design rules	Uni-cell rule set
Products to be produced	Five products, each three to five production steps
Distinct production technologies	Five different process types
Scaling factors	fSF=1; cSF: n.a.
Number of distinct solutions possible (TCS)	52 system configurations
Number of solutions synthesized	250 system configurations (5 sets with 50 solutions each)
Number of distinct solutions synthesized (CCS)	49 system configurations

produced in its own cell). Due to the different ways of allocating the five products to a maximum of five cells, 52 distinct ways for partitioning the products are possible.

The result of generating 250 system configurations with the uni-cell rule set were 49 distinct solutions. Figure 47 (top) shows that allocations to one single cell (ID₁) and to five individual cells (ID₄₉) are more frequent than the other configurations. This is due to the fact that the one- and five-cell configurations imply only one opportunity for allocating the products and expresses the higher synthesis probability ($p=0,2$ for each configuration type in each synthesis attempt; the readers should note that the y-axis does not show the full scale to facilitate the visual comparison). Other configurations are instantiated multiple times and few configurations only once.

Figure 46 shows the results of the performance analysis of the sets. The results show an interesting pattern that is different compared to experiment 3: all sets show the same minimum and maximum values for the investment and utilization. The cause for this phenomenon is the system configurations with one- and five-cells, which are contained in all solution sets due to their higher synthesis probability. While the five-cell configurations require the highest investment, they yield the lowest utilization. The other performance extrema are due to the one-cell configurations which require the least investment but have the highest utilization. Based on this finding, it can be inferred that the solutions with two to four cells lie between these minimum and maximum values, representing various opportunities for decision-makers to choose from based on their preferences.

Figure 46: Maximum (left) and minimum property values (right) observed in the synthesis runs (uni-cell rule set)

Experiment 5: Solution variety of the multi-cell rule set

Compared to the other experiments, an additional constraint is formulated for synthesizing the multi-cell configurations (see Table 11). Specifically, the total number of FAGs in the system configurations is set to $nFAG=5$ to be able to comprehensively itemize the solution space. As the multi-cell rule set is used, the resulting configurations can each have between one to four cells for grouping the four distinct FAGs and one second instance of one FAG type. This yields a TCS of 140 unique hardware configurations [117].

Figure 47: Observation frequency of the distinct system configurations of the uni-cell and multi-cell rule sets in the TCS (x-axis) and CCS (bars) in E4 (top graph) and E5 (bottom graph)

Table 11: Profile of experiment 5

<i>Objective of experiment</i>	Variety of the multi-cell configurations (product and FAG allocation; performances)
<i>Design rules</i>	Multi-cell rule set; number of FAGs nFAG= 5
<i>Products to be produced</i>	Four products, each two to four production steps
<i>Distinct production technologies</i>	Four different process types
<i>Scaling factors</i>	fSF=2; cSF: n.a.
<i>Number of distinct solutions possible (TCS)</i>	140 system configurations
<i>Number of solutions synthesized</i>	250 system configurations (5 sets with 50 solutions each)
<i>Number of distinct solutions synthesized (CCS)</i>	99 system configurations

The 250 generated system configurations resulted in 99 distinct ones. The four possible one-cell configurations (configuration IDs 1 to 4 in bottom graph in Figure 47) are found more often compared to the configurations with more than one cell. Among the other configurations, some are synthesized once, others multiple times and 41 solutions are not synthesized. While the 99 distinct configurations already represent a large variety, the potential performances of the 41 solutions that were not contained in any set can be interpreted as a reason to perform further runs for yielding additional configurations.

Also for this experiment the performance of the configurations is analyzed (cf. Figure 48). Similar to Experiment 3, the resulting minima for investment and utilization differ between the sets, yet are comparable in dimension. At the same time, all sets contain the maximum value utilization of approximately 18,4%, and two distinct maximum values for the investment. The cause for these minimum and maximum values is the four-cell configurations that require the highest investments, which can differ due to various machine duplicates and their associated cost.

Figure 48: Maximum and minimum property values observed in the synthesis runs (multi-cell rule set)

5.2.3 Discussion

The design properties examined in experiments 1 and 2 – the number of cells and the allocation of FAGs – exhibit patterns caused by the bounded randomization. While the arithmetic means and standard deviations of solution properties can be roughly predicted, the occurrence frequency of the possible values of the examined design variables cannot be anticipated due to the semi-random synthesis process. Furthermore, the results of the concatenated randomization assessed in the experiments 3 to 5 and presented in Table 12, show that an exhaustive generation cannot be expected even if the number of generated solutions is considerably higher than the number of distinct configurations. Some configurations in each experiment were synthesized multiple times, which can indicate that they have higher synthesis probabilities. While others were instantiated only once, a varying number of solutions was not synthesized at all in the experiments 2 to 4. For the smaller design spaces (1, 4) an almost exhaustive generation could be achieved, whereas the coverage in larger solution spaces was less (2,3,5).

Table 12: Results of experiments 1 to 5

Experiment	Generated solutions	Distinct solutions generated (CCS)	Distinct solutions possible (TCS)	Coverage (CCS/TCS)
1	50	3	3	100,0%
2	50	43	128	33,6%
3	250	109	128	85,2%
4	250	49	52	94,2%
5	250	99	140	70,7%

Even though these statistics are informative, the readers should note that they cannot be determined unless the number of configurations in the TCS is known. The size of the TCS strongly depends on the problem context and might be difficult to determine. For instance, the two user-specified scaling factors (cSF and fSF) and the distinct machine types required for each product family are the decisive factors for the product-family rule set. Setting both factors to $cSF = fSF = 2$ already leads to a significant increase of the number of solutions, resulting in 60.544 potential system configurations if no requirements apply (see Appendix A.10). However, in case requirements apply to the solutions, the share of the TCS that becomes infeasible depends on the specific solution properties, making it difficult for the users to know how many of their actual options the DSET actually suggested.

A potential drawback of the DSET becoming obvious in the experiments is that the same solution for a specific problem can be generated multiple times and the duplicates do not add to the variety of presented solutions. One way to avoid this effect is to add procedures that check solutions before they are added to the final set, discarding solutions if they are already present in the CCS. Furthermore, sampling strategies different from the semi-random synthesis could be implemented, such as Latin Hypercube Samples which are used in other DSE applications [95]. Alternatively, in case the problem size allows to enumerate the TCS in reasonable time, all feasible candidates and their performances could be stored in a repository, creating the CCS by randomly querying this repository. Instead of generating new solutions when additional requirements are imposed, the queries are

checked. If some solutions turn out infeasible, a selection of other feasible solutions can be displayed in their place. In this way, it could be made sure that no solutions that actually match the users' requirements are concealed, e.g. due to a low synthesis probability or too few generated solutions. While this could allow to expand the variety of solutions in the CCS even further, a sufficiently fast generation of the TCS would need to be ensured despite its varying and potentially enormous dimensions.

5.3 IMPACT OF CHANGED USER SPECIFICATIONS

Next to the coverage of the solutions, the problem- and constraint-focused rationale of the design procedures make the formulation of problems and requirements crucial activities that can have decisive impact on the resulting solutions. The following sections examine the effects of varying the user input on the output of the design procedures. Experiments were designed to determine the consequences of varied problem information (experiment 6), varied design requirements (experiment 7) and varied requirements for the technical performance of the system configurations (experiment 8) and are described in the following sections. Eventually, the findings and implications are discussed. At the start of each experiment, the system configurations synthesized for an initial input data set are considered. After that, the input data is changed and new solutions generated. By comparing the solutions of the original and the changed data set, the impact of the change is determined.

5.3.1 Varying problem information

The problem information of experiment 6 is presented in Table 13. In the varied case, the duration of the second production step of product Po2 is longer than in the original case, representing an higher assumed processing requirement for the production step (see Table 14).

Table 13: Profile of experiment 6

<i>Objective of experiment</i>	Impact of varied problem information
<i>Design rules</i>	Product-family rule set
<i>Products to be produced</i>	Two families with two products, each two to four production steps
<i>Distinct production technologies</i>	Four different process types
<i>Scaling factors</i>	cSF=1 ; fSF=1
<i>Difference between original case and variation</i>	One production step longer than in original case
<i>Number of solutions to be generated by the DSET</i>	20 system configurations
<i>Comparison of results</i>	Original: 20 total, 1 distinct Variation: 20 total, 13 distinct

The solutions obtained for the original data consist of twenty system configurations, all representing the same configuration type (see Table 15) since the scaling factors left no opportunities for randomization in the product-family rule set. The result of the varied case, however, were a total of thirteen distinct configurations. The root cause for this increase of variety can be detected by considering the number of FAGs instantiated in Cell 1 of the respective configurations: the increased processing requirement leads to an increase of the

Table 14: Original and varied input data of experiment 6

Prod./ Prod. fam.	Demand volumes Y ₀ ; Y ₁ ; ...	Original data	Varied data
P01/ 1	19.200; 15.360; 12.288; 9.831; 7.865; 6.292; 5.034; 4.028; 3.223; 2.579	FAG ₁ - 18,0 - 1, FAG ₂ - 60,0 - 1 , FAG ₁ - 10,0 - 1, FAG ₄ - 18,0 - 3	<i>same as original</i>
P02/ 1	20.000; 16.000; 12.800; 10.240; 8.192; 6.554; 5.244; 4.196; 3.357; 2.686	FAG ₂ - 36,0 - 2, FAG₁ - 16,0 - 3 , FAG ₃ - 14,0 - 3, FAG ₁ - 18,0 - 3	2nd step: FAG₁ - 182,0 - 3
P03/ 2	3.800; 3.040; 2.432; 1.946; 1.557; 1.246; 997; 798; 639; 512	FAG ₃ - 10,0 - 3, FAG ₁ - 9,0 - 1	<i>same as original</i>
P04/ 2	5.300; 4.240; 3.392; 2.714; 2.172; 1.738; 1.391; 1.113; 891; 713	FAG ₁ - 7,0 - 3, FAG ₃ - 12,0 - 1, FAG ₂ - 14,0 - 2	<i>same as original</i>

minimum capacity requirements, affecting the minimum number of required FAGs. As a result, multiple parameter values are possible, leading to a larger design space. Table 15 presents the configuration with the fewest and the most equipment instances created for the varied problem data (solutions 1 and 13): the same utilization rate of both solutions shows that including a third FAG instance of any type does not make the production process faster. This is because the analysis model (Appendix A.8) implies that only two FAG instances can work on the same product at the same time. While the additional machines are unused, they increase the cost of the cell and create a difference between the direct production cost of both configurations, also in relation to the original cost.

Compared to the configuration resulting from the original case, the new configurations exhibited higher utilization and direct production cost, both due to more machines that have to be programmed and changed-over for production. Yet, no general rule can be formulated since the factor for the number of FAG instances could remain the same based on different problem data (e.g. higher FAG speed). In this case, the cell utilization would increase compared to the original case. This shows that the effects of changing the problem

Table 15: Resulting system configurations in the original (left) and varied case (right)

	Original case solution		Varied case solution 1 (minimum equipment)		Varied case solution 13 (maximum equipment)	
	Cell 1 PF 1	Cell 2 PF 2	Cell 1 PF 1	Cell 2 PF 2	Cell 1 PF 1	Cell 2 PF 2
System hardware configuration	FAG ₁ (1x)	FAG ₁ (1x)	FAG ₁ (2x)	FAG ₁ (1x)	FAG ₁ (3x)	FAG ₁ (1x)
	FAG ₂ (1x)	FAG ₂ (1x)	FAG ₂ (2x)	FAG ₂ (1x)	FAG ₂ (3x)	FAG ₂ (1x)
	FAG ₃ (1x)	FAG ₃ (1x)	FAG ₃ (2x)	FAG ₃ (1x)	FAG ₃ (3x)	FAG ₃ (1x)
	FAG ₄ (1x)		FAG ₄ (2x)		FAG ₄ (3x)	
	CellBase: 1 CU, 1 TT 1 Rob, 11 Tools	CellBase: 1 CU, 1 TT, 1 Rob, 6 Tools	CellBase: 1 CU, 1 TT, 1 Rob, 11 Tools	CellBase: 1 CU, 1 TT, 1 Rob, 6 Tools	CellBase: 1 CU, 1 TT, 1 Rob, 11 Tools	CellBase: 1 CU, 1 TT, 1 Rob, 6 Tools
Cell utilization	30,3%	4,0%	36,3%	4,0%	36,3%	4,0%
Direct production cost	228.133 €		524.697 €		719.835 €	

specification strongly depend on the case-context and can have considerable effects on the number of the synthesized configurations, their design and performance, even without changing the scaling factors or requirements.

5.3.2 Varying design requirements

The data of experiment 7 is presented in Table 16. An embodiment requirement limits the shop floor area of the configurations, specifying a maximum of 130m² for the original case and 120m² in the variation.

Table 16: Profile of experiment 7

<i>Objective of experiment</i>	Impact of varied design requirements
<i>Design rules</i>	Uni-cell rule set
<i>Products to be produced</i>	Five products, each three to five production steps
<i>Distinct production technologies</i>	Five different process types
<i>Scaling factors</i>	fSF=1; cSF: n.a.
<i>Original requirements</i>	two production cells or less; less than 130m ² shop-floor occupancy
<i>Varied requirements</i>	two production cells or less; less than 120m ² shop-floor occupancy
<i>Difference between original case and variation</i>	Less shop-floor area consumption allowed in variation
<i>Number of solutions to be generated by the DSET</i>	20 system configurations
<i>Comparison of results</i>	Original: 20 total, 7 distinct Variation: 20 total, 1 distinct

Seven distinct configurations resulted from the original data, (cf. Table 17; duplicates not presented), but all configurations resulting from the varied case were similar to PS_1. Considering the performances of the original results, it can be seen that the variation of the constraint on the space occupancy renders the system configurations PS_2 to PS_7 infeasible. Hence, the variation of the embodiment constraints drastically reduces the number of resulting system configurations and eliminates the variety.

Table 17: Resulting system configurations of the original (PS_1 to PS_7) and the varied case (PS_1)

Sol. ID	System configuration	No. of cells	Area [m ²]
PS_1	Cell_1: P05, P01, P04, P03, P02	1	66
PS_2	Cell_1: P02 — Cell_2: P04, P05, P01, P03	2	120
PS_3	Cell_1: P05, P03, P01 — Cell_2: P04, P02	2	124
PS_4	Cell_1: P03, P01, P02, P04 — Cell_2: P05	2	127
PS_5	Cell_1: P03, P04, P01 — Cell_2: P05, P02	2	127
PS_6	Cell_1: P05, P04 — Cell_2: P03, P02, P01	2	127
PS_7	Cell_1: P01 — Cell_2: P04, P05, P03, P02	2	129

5.3.3 Varying performance requirements

The data for experiment 8 is presented in Table 18. For examining the effect of varied performance requirements, the utilization rate is varied. The performance constraint on the utilization ratio specifies a minimum of 36% in the original case and 39% for the varied case. To allow for duplicates of machines, the fag scaling factor is set to $fSF = 2$. A potentially negative consequence of the multi-cell rule set can be that product flows are often interrupted due to the required transfers of batches between the cells. Since the consequence of this can be high stock levels, a performance requirement specifies the maximum stock levels to emulate a limited storage capacity. The result of the run with the original settings

Table 18: Profile of experiment 8

<i>Objective of experiment</i>	Impact of varied performance requirements
<i>Design rules</i>	Multi-cell rule set
<i>Products to be produced</i>	Four products, each two to four production steps
<i>Distinct production technologies</i>	Four different process types
<i>Scaling factors</i>	$fSF=2$; cSF : n.a.
<i>Original requirements</i>	Stocklevel less than 1000 units; utilization larger than 36%
<i>Varied requirements</i>	Stocklevel less than 1000 units; utilization larger than 39%
<i>Difference between original case and variation</i>	Increased required utilization rate in variation
<i>Number of solutions to be generated by the DSET</i>	100 system configurations
<i>Comparison of results</i>	Original: 33 total, 7 distinct Variation: 4 total, 1 distinct

were thirty-three solutions that consisted of five distinct configuration types (cf. Table 19). The varied case brought a total of four solutions that all embodied the same configuration,

Table 19: Resulting system configurations of the original (PS₉₁₀ to PS₁₄₁₄) and varied case (PS₁₄₁₄)

Name	System hardware design	Stock-level	Utilization
PS ₉₁₀	Cell ₁ : FAG1(2x), FAG2(2x), FAG3(1x), FAG4(2x) CellBase: 1 CU, 1 TT, 1 Rob, 11 Tools;	955	0,363
PS ₉₇₂	Cell ₁ : FAG1(2x), FAG2(2x), FAG3(2x), FAG4(1x) CellBase: 1 CU, 1 TT, 1 Rob, 11 Tools;	955	0,373
PS ₁₀₁₃	Cell ₁ : FAG1(2x), FAG2(1x), FAG3(2x), FAG4(2x) CellBase: 1 CU, 1 TT, 1 Rob, 11 Tools;	955	0,374
PS ₁₀₆₉	Cell ₁ : FAG1(1x), FAG2(2x), FAG3(2x), FAG4(2x) CellBase: 1 CU, 1 TT, 1 Rob, 11 Tools;	955	0,372
PS ₁₄₁₄	Cell ₁ : FAG1(2x), FAG2(2x), FAG3(2x), FAG4(2x) CellBase: 1 CU, 1 TT, 1 Rob, 11 Tools;	955	0,399

similar to PS_1414. An aspect worth noting is that all results represent configurations that produce the products on a single cell, despite the various opportunities of the procedure to allocate each product to multiple cells. The reasons for this is the combination of the requirements on stock-levels and utilization. While the stock-level constraint renders all solutions with inter-cell transfers infeasible, the utilization requirement prevents that the capacity (and its utilization) is distributed over multiple cells, which would make the utilization fall below the specified threshold. As a result, the mere increase of the required utilization rate has eliminated the variety of solutions.

5.3.4 *Discussion*

The previous sections show that already slight changes of the input information can have considerable impact on the resulting number and variety of solutions. Experiment 6 shows that the specification of the problem is very important for the variety of the synthesized configurations due to the automated problem analysis that calculates the possible minimum and maximum values for the number of FAGs. Also, the variation of the values used for embodiment and performance constraints in experiments 7 and 8 influence the number and other characteristics of the resulting solutions. Some of these observed effects could have been anticipated, if the results of the original case had been considered before specifying discrete values for the variation. Yet, the cases show that the parametrization of the problem is crucial when there is no familiarity with the CCS and TCS. This suggests that the users should be aware of their freedom to attribute discrete values for the different parameters, particularly when no or only very few solutions result from specific problem data or constraint values. While the constraints on design properties may be more difficult to negotiate due to brownfield situations on the shop-floor or to maintain a manageable complexity of the product flows, more flexibility seems to be given for constraints on the performances of the system, e.g. to make trade-offs between system utilization, logistics and storage costs. The uncertainty with regard to the future in terms of product demand and design makes the DSET suited to approach various what-if problem specifications in an experimental way. Since the synthesis of solutions was performed in less than two seconds in all described experiments, multiple design problems can be efficiently approached.

Hence, strategically varying the values of both input information and constraints can not only influence the number of available candidate configurations, but can also allow to identify interdependencies between the various characteristics of system configurations. By finding novel solution characteristics after relaxation of constraints or finding characteristics eliminated after applying more rigorous constraints, the users can develop a feeling for the decisive parameters of their problem and the configurations' sensitivity. The large number of parameters that can be specified by the users represents a quality of the DSET that can be considered a drawback and a strength at the same time. On the one hand, it may be difficult to determine which out of the many properties to vary first, when no system configurations can be found for a specific problem formulation and requirements. On the other hand, the DSET's efficiency makes it possible to evaluate the feasibility of alternative sets of problem specifications and constraints, as they can be straightforwardly tested by using different property values.

5.4 COMPARING SOLUTION SPACES OF VARIOUS PROBLEMS

Even though the experiments described in the previous sections examine the variety of the synthesized system configurations with respectively one set of design rules, the rationale of the DSET is to use the three design-rule sets simultaneously to suggest various solutions for the same problem. The effects of this rationale also need to be assessed. This section therefore evaluates the variety of production system configurations obtained in various problem settings. For this purpose, two distinct sets of objectives are defined to serve as narratives for applying the DSET, as shown in Table 20: in experiments 9 and 10, the system configurations should require little investments and provide strategic capacity reserves; different to that, the requirements applied in experiments 11 and 12 specify that the system configurations consist of few cells and cause low total cost. The underlying assumption of these narratives is that the solution space is not known beforehand, hence the design and performance objectives for the system configurations can be only expressed qualitatively and cannot specifically quantified. Additionally, the experiments considered varying numbers of products and process types - five products and five process types for experiment 9 and 10 and twenty-five products and eight process types in experiments 11 and 12. Next to the number of products to be considered, a difference between the categories was that the products in experiments 9 and 10 had sequence constraints for the allocation of the production steps, whereas in experiments 11 and 12 such constraints were not applied. This was to assess the influence of sequence constraints on the number of solutions. The two problem instances considered for the five and twenty-product problems were created with randomized input information, representing unknown design problems to indicate the capability of the procedures to find solutions independently of the specific parametrization of the instance.

Table 20: Evaluation experiments for comparing solution spaces

Exp.	Prob. instance	Objectives	Number of products	Sequence constraints	Number of process types
E9	5P-PI1	Strategic capacity reserve & low investment	5	yes	5
E10	5P-PI2		5	yes	5
E11	25P-PI1	Few cells & low total cost	25	no	8
E12	25P-PI2		25	no	8

For all experiments, particular synthesis strategies were applied. Since the FAG scaling factor fSF and the cell scaling factor cSF strongly affect the synthesis of configurations, suitable values for the synthesis runs had to be determined. While the most suitable solutions for the indicated requirements were expected to be achieved by using low values for these factors, the previous section shows that the exact parametrization for yielding suitable solutions is difficult to know upfront. Various values were therefore established. Table 21 shows the values for the product-family rule set on the left. For each combination of fSF and cSF, a set of 100 system configurations are requested, resulting in 31 runs and a maximum of 3.100 solutions. As the cell scaling factor is not relevant for uni-cell and multi-cell rule sets, only the fag scaling factor is varied. To obtain the same number of solutions as with the product-family rule set, multiple runs were performed with the same fSF value (see Table 21, right). The number of runs performed with the distinct fSF values is the

Table 21: Scaling factors for the product-family rule set (columns 1-6) and the uni-cell and multi-cell rule set (rightmost column)

	Product-family rule set						Multi-cell & uni-cell rule set
	cSF = [1, 10]	cSF = 2,0	cSF = 1,5	cSF = 1,2	cSF = 1,1	cSF = 1,0	
fSF = [1, 10]	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 1-6
fSF = 2,0	X	Run 7	...				Run 7-11
fSF = 1,5	X						Run 12-16
fSF = 1,2	X						Run 17-21
fSF = 1,1	X						Run 22-26
fSF = 1,0	X			...	Run 30	Run 31	Run 27-31

identical for all rule sets. As a result, a total of 93 runs and a maximum of 9.300 solutions were performed for each experiment.

To face the high number of solutions in each experiment and integrate the objectives for the solutions, two forms of selection strategies were applied to select suitable candidates:

- *Pareto*: The first strategy is based on the feature for visualizing the solutions on the Pareto-fronts for the indicated requirements, selecting only the Pareto-efficient solutions of all runs for each rule set.
- *Manual*: The second strategy consists of selecting suitable solutions based on a visual comparison of the target parameters.

The motivation is to assess whether the solutions selected based on the visual comparison are vastly different to the solutions resulting from the Pareto-strategy. In the following, Section 5.4.1 presents the results of the experiments 9 and 10 considering five products. Section 5.4.2 presents the outcomes of the experiments 11 and 12 with twenty five products. Eventually, Section 5.4.3 discusses the results.

5.4.1 System configurations with strategic capacity reserve

Table 22 presents the information of the experiments performed based on the first problem instance.

Experiment 9: Five-product-case instance 1 (5P-PI1)

The total number of solutions generated (7.919 solutions) differed from the requested number by 1.381 solutions. The product-family and uni-cell rule sets synthesized the requested number of solutions, yet, the multi-cell rule set failed to achieve the requested number. The reason for this is that the hardware configurations are created first and the products allocated afterwards. As the problem formulation included sequence constraints for the products, it is possible that the suggested hardware configurations prove infeasible for products with such constraints, resulting in the discard of the solution. In case this occurs repeatedly, the maximum number of synthesis attempts is reached and the synthesis run is aborted, leading to fewer solutions than requested. The resulting system configurations are represented by their performances for the Pareto selection strategy in Figure 49 and the manual selection in Figure 50. The Pareto-efficient system configurations suggest that in general, the lowest investment for a specific utilization rate can be reached with the multi-cell rule

Table 22: Profile of experiment 9 (5P-PI1)

<i>Objective of experiment</i>	Configurations with strategic capacity reserve and low investment
<i>Design rules</i>	All rule sets (product-family; uni-cell; multi-cell)
<i>Products to be produced</i>	Five products, each two to six production steps
<i>Distinct production technologies</i>	Five different process types
<i>Scaling factors</i>	see Table 21
<i>Number of solutions to be generated by the DSET</i>	Total: 9.300 solutions (31 synthesis runs of 100 solutions with each of the three rule sets)
<i>Number of solutions synthesized</i>	Total: 7.919 (product-family: 3.100; uni-cell: 3.100; multi-cell: 1.719)

set, followed by the uni-cell configurations and eventually the product-family configurations. Also in absolute terms, the solutions with the highest utilization rates are multi-cell configurations, which represent the only configuration type with capacity utilization rates of more than 50% and simultaneously requiring the least investments. Also considering the manually selected configurations, the ratio of investment to utilization is better for the multi-cell configurations. Moreover, the manual selection strategy shows many uni-cell configurations with a characteristic utilization level of approximately 45%, where only the investments seem to vary. By assessing the details of these solutions, one can determine that all these configurations only consist of one cell (with 45% utilization), while the different investment values are caused by varying numbers of redundant equipment instances. As the investment required for these system configurations is too high, they do not appear in the Pareto-efficient set.

The root cause for observing cheaper and more utilized multi-cell configurations is the freedom of allocating individual production steps on different cells, requiring less equipment in the total system. This makes possible solutions with low investments and allows for increasing the utilization of these fewer FAGs. Considering the requirements of the case allows to conclude that if the strategic capacity reserve should lie between 20% and 40%, the most suitable system configurations could be chosen among the multi-cell solutions. In case an even higher reserve is needed, all types of system configurations found in the lower half of the graphs represent potential candidates.

Figure 49: Investment and utilization of the Pareto-selected solutions (5P-PI1)

Figure 50: Investment and utilization of the manually selected solutions (5P-PI1)

Experiment 10: Five-product case instance 2 (5P-PI2)

The profile of the second experiment with five products is presented in Table 23.

Table 23: Profile of experiment 10 (5P-PI2)

<i>Objective of experiment</i>	Configurations with strategic capacity reserve and low investment
<i>Design rules</i>	All rule sets (product-family; uni-cell; multi-cell)
<i>Products to be produced</i>	Five products, each two to six production steps
<i>Distinct production technologies</i>	Five different process types
<i>Scaling factors</i>	see Table 21
<i>Number of solutions to be generated by the DSET</i>	Total: 9.300 solutions (31 synthesis runs of 100 solutions with each of the three rule sets)
<i>Number of solutions synthesized</i>	Total: 8.252 (product-family: 3.100; uni-cell: 3.100; multi-cell: 2.052)

Also in the second five-product case, the number of requested solutions was not fully generated due to the multi-cell rule set, showing a difference of 1.048 configurations. With regard to the Pareto-efficient solutions (cf. Figure 51), the three different configuration types yield more homogenous performances than observed in the previous experiment. The configurations resulting from both the multi-cell and uni-cell rule sets perform similarly in the range between approximately 28% and 46% capacity utilization and the identical performance values for these configurations suggest that the rule sets led to the same solutions. In the range between 17,5% and 22,5% capacity utilization all configuration types are present. Less distinctive performances of the three configuration types compared to the previous experiment can be also observed among the manually selected candidates, however, also here the multi-cell and uni-cell configurations allow for the highest utilization rates (cf. Figure 52).

In the considered case, the highest utilization levels are achievable with the uni-cell and multi-cell rule sets with maximum values around 45%. Less utilization and larger capacity reserves can be achieved by investing only slightly more. In contrast to the previous case instance there is not an individual configuration type that proves most suitable for the indicated requirements, but both the uni-cell and multi-cell rule sets yield equally well-performing solutions. Assessing the input data allows to detect the cause for this phenomenon: the sequence constraints of product P03 lead to the infeasibility of all configurations whose production resources are distributed across multiple cells. Due to the targeted low investments, it is impossible to find a feasible configuration with minimum equipment that is distributed across multiple cells. Another aspect worth noting is that the investments required for the product-family configurations are closer to multi-cell and uni-cell solutions than in the previous instance. Depending on the envisaged level of capacity reserve and willingness to invest, these configurations could be suitable alternatives.

Figure 51: Investment and utilization of the Pareto-selected solutions (5P-PI2)

Figure 52: Investment and utilization of the manually selected solutions (5P-PI2)

5.4.2 System configurations with low cost

The objective for the experiments with twenty-five products is to find system configurations with low total cost, simultaneously consisting of few cells. To increase the chance of obtaining feasible multi-cell configurations, no sequence constraints are specified for the products.

Experiment 11: Twenty-five-product-case instance 1 (25P-PI1)

Table 24: Profile of experiment 11 (25P-PI1)

<i>Objective of experiment</i>	Configurations with low total cost and few cells
<i>Design rules</i>	All rule sets (product-family; uni-cell; multi-cell)
<i>Products to be produced</i>	Twenty-five products, each two to six production steps
<i>Distinct production technologies</i>	Eight different process types
<i>Scaling factors</i>	see Table 21
<i>Number of solutions to be generated by the DSET</i>	Total: 9,300 solutions (31 synthesis runs of 100 solutions with each of the three rule sets)
<i>Number of solutions synthesized</i>	Total: 9,300 (product-family: 3,100; uni-cell: 3,100; multi-cell: 3,100)

In the first twenty-five product experiment, the full number of requested solutions was generated. Figure 53 presents the Pareto-efficient system configurations. While the total cost throughout the years seem to be comparable for the product-family and uni-cell configurations, the multi-cell solutions expose significantly higher cost. Moreover, the system configurations with the lowest number of cells are the uni-cell solutions. The same impression arises when considering the manual selection of the most suitable configurations presented in Figure 54. Additional uni-cell configurations with three and four cells are present, which are not contained in the Pareto-selection set. Also additional product-family configurations with six cells can be found in the manual selection.

For both manual and Pareto-based selections, the total cost of the solution types differ drastically. Examining the cause for the high total cost of the multi-cell configurations shows that while they can be realized with comparably low initial investment, the logistics cost throughout the operation horizon of the configurations are significantly higher with both selection strategies (cf. Figure 55 and Figure 56). This emphasizes the importance of considering the entire system-lifecycle, where the monetary effects of separating the production steps – inducing the higher logistics cost – outweigh the benefits of initially requiring less production equipment and investments.

Figure 53: Number of cells and total cost of the Pareto-selected solutions (25P-PI1)

Figure 54: Number of cells and total cost of the manually selected solutions (25P-PI1)

Figure 55: Investment and logistics cost of the Pareto-selected solutions (25P-PI1)

Figure 56: Investment and logistics cost of the manually selected solutions (25P-PI1)

Experiment 12 - twenty-five-product-case instance 2 (25P-PI2)

The details of the second experiment with twenty-five products are presented in Table 25.

Table 25: Profile of experiment 12 (25P-PI2)

<i>Objective of experiment</i>	Configurations with low total cost and few cells
<i>Design rules</i>	All rule sets (product-family; uni-cell; multi-cell)
<i>Products to be produced</i>	Twenty-five products, each two to six production steps
<i>Distinct production technologies</i>	Eight different process types
<i>Scaling factors</i>	see Table 21
<i>Number of solutions to be generated by the DSET</i>	Total: 9.300 solutions (31 synthesis runs of 100 solutions with each of the three rule sets)
<i>Number of solutions synthesized</i>	Total: 9.300 (product-family: 3.100; uni-cell: 3.100; multi-cell: 3.100)

Also for this experiment, the requested number of solutions was generated by the algorithms. Figure 57 shows that only one multi-cell configuration is part of the Pareto-efficient solutions, which again induces higher total cost than the uni-cell and product-family configurations. Moreover, while there appears to be a difference with regard to number of cells between the Pareto-efficient solutions of the uni-cell and product-family configurations, many configurations were found with comparable total cost. The manual selection shown in Figure 58 allows to detect additional uni-cell configurations with more than three cells. Yet, it conveys a similar impression as in the previous experiment, where the multi-cell configurations induce higher total cost.

To investigate the reasons for that, the logistics cost of the Pareto-selected set (cf. Figure 59) are considered. The single multi-cell configuration in the set induces the same logistics cost as the other configuration types. Analyzing the other cost-related performances of the multi-cell solution shows that the difference is due to higher direct production cost and higher utilization rates. The manually selected configurations (cf. Figure 60) show that the higher logistics cost of the multi-cell configurations is the reason for the higher total cost of the multi-cell configurations in this set. Hence, it can be concluded that the uni-cell candidates that result from the Pareto-selection strategy can be considered viable candidates for the indicated requirements.

Figure 57: Investment and logistics cost of the Pareto-selected solutions (25P-PI2)

Figure 58: Investment and utilization of the manually selected solutions (25P-PI2)

Figure 59: Investment and logistics cost of the Pareto-selected solutions (25P-PI2)

Figure 60: Investment and logistics cost of the manually selected solutions (25P-PI2)

5.4.3 Discussion

The previous sections assess the effects of using the DSET in various problem situations, embodied by different requirements, problem sizes and problem instances. Multiple and various system configurations were found for all problem specifications.

In experiments 9 and 10, less solutions could be obtained with the multi-cell rule set, whereas in experiments 11 and 12, the full number of requested solutions was generated. The causal difference between these cases is that sequence constraints apply to the products of experiments 9 and 10, whereas such constraints do not apply to the products in experiments E11 and E12. Even though experiments 11 and 12 imply a much larger number of products to be allocated to the randomly generated hardware configurations, the requested number of solutions could be obtained. This suggests that the multi-cell rule set is sensitive to the sequence constraints.

Regarding the variety of the solutions, each experiment allows to observe partly overlapping performance values of the various solution types, such as the region with investment $[0, 1.000.000]$ and utilization $[0, 0,25]$ in Figure 49. In case such performances reflect the decision-makers' requirements and objectives, all those system configurations represent suitable candidates. Moreover, the different design-rule sets open up distinctive regions in the solution space, for example the region with investment $[0, 1.000.000]$ and utilization $[0,5, 1]$ for multi-cell configurations in Figure 49. Hence, no general statement can be made about which design-rule set provides the most useful system configurations. Instead, the exact formulation of the problem in combination with the users' requirements and objectives determine the suitability and relevance of the proposed system configurations and configuration types. Similar statements can be made with regard to the synthesis and selection strategies. Testing various requirement combinations – such as the fSF and cSF combinations in the previous sections – can enable to find solutions with more viable characteristics, when compared to synthesizing sets with the default settings. Yet the effect of setting particular requirements are unique to the problem instance and call for an experimental attitude to their formulation.

Moreover, the different features of the system configurations resulting from the Pareto and manual selection strategies highlight the importance of being aware of the selection strategies' implications. In experiment 11, for example, not a single configuration with three production cells results from the Pareto-selection, whereas the manual selection included solutions with this property value. While the DSET makes it possible to detect such differences – which can be interpreted as expression of its versatility – it also implies that the users should be aware of the implications of this mathematical approach to describing their preference.

Similar to the experiments in Section 5.2.1, the generated sets contained duplicate system configurations, which means that the number of solutions generated for each case does not quantify the variety of configurations. Yet, the figures (Figure 49 to 60) visualize various performance values that are due design differences of the system configurations, which means that generating large varieties of solutions is possible.

The experiments demonstrate only a few out of the many ways in which the DSET can be used to find various solutions for different problems and requirements. The synthesis procedure required less than five seconds per requested hundred solutions, thereby contributing to efficiently considering a large number of system configurations. The customizability of the DSET – provided by the synthesis procedures and the GUIs – makes it possible to

examine related problem formulations by combining various synthesis strategies, sets of requirements and selection strategies.

5.5 CONCLUSION

The previous sections describe the evaluation of the DSET with regard to the internal specifics and resulting implications for the users, answering RQ3: *What are the characteristics of the suggested approach to design automation?*. It is shown that the DSET is capable of generating a variety of production system configurations by assigning system design properties semi-randomly. In this context, the number of solutions that match the indicated requirements and their synthesis probability are identified as decisive factors for obtaining a large variety of configurations, in combination with the number of solutions requested by the users. Moreover, it was shown that the problem formulation is an important factor regarding the resulting system configurations. The users implicitly determine the possible size of the resulting solution spaces by specifying the problem information (such as the expected demand) and setting the scaling factors, whereas entering the requirements can reduce these spaces. It is possible that the users' requirements render all potential solutions infeasible. As decision-makers are mostly unfamiliar with the solution space associated to their problems, all input data and requirements should be assessed for applicable tolerances. If there are opportunities for varying values, various synthesis runs can be performed to strategically test sets of problem information, scaling factors and requirements. In this manner, the sensitivity and interdependencies of design and performance aspects of the solutions can be detected. Using the design procedures to generate solutions and using the GUI to evaluate and select the most suitable configurations enables the users to efficiently assess and explore a variety of system configurations. Also, custom selection strategies can be applied. Since the problem data, the requirements and objectives can be easily changed, the DSET allows to assess the solutions for various kinds of design problems.

These findings allow to formulate recommendations for users of the approach. Firstly, the degrees of freedom should be identified with regard to formulating the problem, embodiment and performance requirements. Secondly, experiments can be designed in the fashion of factorial experiments, varying all parameters and one factor at a time (as shown in Section 5.3), including also a variation of the scaling factors (similar to Section 5.4). After generating solutions, various selection strategies should be applied to select the most suitable solutions from the results of all experiments. Eventually, these most suitable solutions should be assessed individually, including a critical review of the specific configuration, the associated performances, underlying assumptions about the problem and implications for reconfiguration. From a technical perspective, three features of the DSET are particularly important in this context: (i) covering a vast number of the real system configuration opportunities by providing suitable design-rule sets; (ii) ensuring the feasibility of the configurations by allowing to vary the formulation of problems and requirements; (iii) and facilitating visual comparisons and selection of the system configurations. This combination should enable the users to find system configurations in varying problem contexts without the need to adapt the underlying code and models, providing effectiveness and efficiency when exploring the broad range of potential system designs.

EVALUATION OF THE DSET IN PRACTICE

This chapter provides an answer to RQ4 (*How applicable and useful is the suggested approach to design automation in practice?*) and describes evaluation of the DSET in a realistic setting. For this purpose, the scope of consideration is broadened and considers the factors described in the previous chapter and, in addition, the context in which the tool is supposed to be applied, i.e. the real design space and the final solutions selected from the CCS (cf. Figure 41). Section 6.1 describes the design of the empirical study. Section 6.2 describes the results, which are discussed in Section 6.3, before drawing the conclusions in Section 6.4.

6.1 RESEARCH APPROACH

The DRM advises to evaluate design support tools in the industrial problem domain so that the tool's applicability and usefulness can be determined by the actual problem owners and to create opportunities for feedback for improving the tool [45]. As the model of the production system implemented in the DSET is based on the system design problem of the company in the Robust PlaNet project use-case, the evaluation could be carried out in collaboration with their experts and represents a holistic-single case research approach [118]. Two workshops were held with the experts, one for letting the experts use the DSET to solve system design problems and answering evaluation questions, and another workshop for validating the findings. In those workshops, semi-structured interviews served as the main source of evidence, since they allow researchers to ask questions of both exploratory and confirmatory nature [119]. The course of these sessions and the related preparation, analysis and appraisal activities, are presented in Figure 61 and explained in the following sections.

6.1.1 Preparation of the experimental session

The first session had the objective to let the participants evaluate the DSET with respect to its capabilities and irrespective of its compatibility to their established processes. A short narrative therefore described the frame and objectives of the session (cf. Appendix A.11). Moreover, two usage scenarios were developed, one representing a configuration problem considering just a few products and one with many products as shown in Figure 62 (see also Appendix A.12). Another difference between these configuration problems were the requirements for the resulting system configurations. For each problem, the engineers were asked to propose a minimum of three alternative system configurations that are suitable for producing the case-specific input set of products. The embodiment requirements – such as constraints on available shop-floor space or for the number of cells – are supposed to emulate the brownfield situation that the engineers face when performing partial reconfig-

Figure 61: Phases and main activities of the practical evaluation

urations of the shop floor. The requirements for the systems' performance are supposed to simulate potential production strategy objectives.

To confirm whether the participants could identify the individual features of the DSET as well as their combined effects, two sets of questions were developed to be asked after each problem case (cf. Appendix A.13): one for evaluating the individual features after solving

Figure 62: Input information for the cases of the empirical evaluation

the first configuration problem, and one for assessing the combined effects after the second configuration problem. These questions were derived from the reference model and aimed at confirming the presence and influence of the support features and their resulting effects (Figure 42). Lastly, two other questionnaires were designed to let the participants evaluate the course of the session and its realism. The prior represents a typical approach to let the participants cool down at the end of the session and give the opportunity to state impressions about its course. The latter was supposed to be asked during the second meeting to indicate if the solved cases are representative for the company's situation, indicating the external validity of the session.

Following the described plan, a pilot experiment was conducted upfront to have the opportunity to improve the evaluation design and to ensure that the materials and questions are complete and can be understood. By choosing a person with practical experience in manufacturing companies and no prior knowledge of the details of the DSET, the pilot experiment aimed at delivering useful feedback and improving the evaluation set-up. This pilot experiment was conducted on the 11th of November 2016.

6.1.2 *Experimental session (data collection I)*

The experimental session was held on the 17th of November 2016 on the company's premises. Two engineers participated in this session: the lead assembly system engineer responsible for designing the production cells, in the following referred to as the designer, and the lead assembly system engineer responsible for implementing the selected system concepts, the implementer. The designer had a total of sixteen years working experience in this job and the implementer two years. Before the session, the participants only had a rough idea about the objectives of the DSET, however they had no experience in operating it or knowledge about the detailed features of the algorithms and the model.

At the beginning, they were briefed about the course of the workshop. Then, the GUIs and features of the DSET were explained to enable the participants to use the tool. Following that, they were presented with the fictive situational context (Appendix A.11) and introduced to the first configuration problem they were supposed to solve, represented by Scenario A and its additions (cf. Appendix A.12). After successfully using the tool to find system configurations for the problem specifications, the first semi-structured interview (cf. Appendix A.13) was conducted to evaluate the influence of the features implemented in the DSET and to document additional observations or impressions of the participants. Subsequently, the second case was presented in the form of Scenario B and successfully solved by the participants. The following semi-structured interview addressed the second set of questions and was answered only by the implementer, as the designer had to leave due to an urgent appointment. Both the designer and implementer were again present during the evaluation of the experimental session at the end of the meeting.

6.1.3 *Data analysis I and preparation of the verification session*

As the first step of data analysis, the audio recording of the first session was transcribed in ATLAS.ti (version 7.5.10) [120], a software tool specifically supporting qualitative data analysis. After finishing the transcription, the text was analyzed and categories – so called codes – assigned to specific text fragments, relating to the subject of the evaluation questions. In this manner, the drawn conclusions can be traced back to the respective text

fragments in the audio recording and in the transcript, facilitating to navigate between statements that address the same aspect and thereby help to avoid misinterpretation. The codes for classifying the feedback of the experimental session were defined as:

- Feature evaluation: text fragments evaluating the DSET's individual features;
- Effect evaluation: text fragments evaluating the combined effects of the DSET's features;
- As-Is process description: text fragments describing the current routines and processes at the company;
- Tool description: text fragments containing subjective remarks and opinions about the DSET and its features;
- Session related comments: text fragments containing subjective remarks and opinions about the meeting;

In the course of assigning the codes, all evaluation questions could be linked to the text fragments presenting the respective answers. For each question, the assigned answers were checked for consistency, summarized and compared to the intended effect model resembled by Figure 42. Based on this outcome, the verification session was prepared to confirm the answers that were found consistent and to address the ambiguous, unclear or unexpected answers.

6.1.4 *Verification and validation session (data collection II)*

The verification and validation session was held on the 24th of November 2016 on the company's premises, one week after the experimental session. The designer and the implementer also took part in the second round. Additionally, a senior manager attended the meeting who was familiar with the rationale of the DSET through the previous master assignments he had co-supervised (referring to [115, 116]). His operative responsibilities at the company spanned the design and management of the entire process chain of the production department with a working experience of twenty-two years.

At the start of the session the participants were given an overview of the results of the experimental session. By explaining the differences between the observed use of the tool and the originally intended effects, the participants could provide their feedback regarding these aspects. Then, the detailed evaluation questions and answers given throughout the first session were presented and compared to the underlying features of the DSET and their originally intended objectives. After that, unclarified aspects and ambiguous answers from the previous session were presented, aiming to let the participants clarify the meaning of their statements. At the end, the participants were asked to evaluate the availability of the input data and the realism of the cases solved during the first session.

6.1.5 *Data analysis II*

Also the audio recording of the second meeting with the company representatives was transcribed in the data analysis software and codes were assigned. The code categories applied for this session were as follows:

- Validated features: text fragments confirming particular findings from the experimental session;
- Availability of input data: text fragments describing the availability of input data;
- Scenario realism: text fragments evaluating the realism of the configuration problems of the first case;
- As-is process description: text fragments describing the current routines and processes at the company;
- Lacking model aspects and tool features: text fragments regarding potential additions to the model and tool features;
- Limited ability to evaluate: text fragments stating a limited ability to evaluate particular model aspects or tool features;

Eventually, a report that summarized all evaluated aspects and findings of the two sessions was created and sent to the sessions' participants for review and approval to ensure the validity of this summary.

6.2 RESULTS

In the following sections, the results of the evaluation study are presented: Section 6.2.1 discusses the evaluated key features of the DSET, Section 6.2.2 the expected higher-level effects and Section 6.2.3 the realism.

6.2.1 *Results of evaluating the key features*

All individual features of the DSET were recognized and confirmed by the participants of the experimental session. The participants found system configurations that differed in the arrangement of production hardware and in product allocation types (Appendix A.13, Q1a, 1b). The automated proposal was found to increase the number of configurations considered, as well as increasing the speed of synthesis and analysis (Q6, 7). Addressing the feasibility of configurations as potential candidates for implementation, the feature of the DSET to specify requirements for particular design or performance properties was confirmed, yet only after explaining the purpose and importance of the requirements during the verification session (Q2a, 2b). The participants could also confirm the features to evaluate the quality of the proposed configurations by assessing the performance sensitivity of configurations in different demand scenarios (Q3); by comparing the system configurations to each other with regard to the various properties (Q4); by identifying promising configurations using the Pareto-front feature and the visual guidance (Q5). Also, they found the approach allowed to easily formulate and change the problem specifications by modifying the input files and using the GUIs to integrate the various requirements and objectives (Q8).

6.2.2 *Results of evaluating the complementary effects*

After the second experimental case was solved by the participants, the higher-level objectives of efficiency and effectiveness – aimed for by combining the effects of the individual

features – were assessed. For this purpose, the individual features previously confirmed by the participants were put into the context of the main objective of the DSET, with Q9 to 14 addressing its effectiveness and Q15 to 19 the efficiency. The participants found that the tool enabled them to assess a considerably larger variety of configurations than they used to consider in their manual process, resulting in a more exhaustive procedure (Q20). Appendix A.14 describes the performance of the solutions that were generated and selected by the participants for Scenario A and its additions, demonstrating their ability to use the DSET. Yet, while discussing which system configuration types were possible additionally to the ones implemented in the DSET, the participants stated that even more types of configurations were possible, hence confirming and relativizing the concept of exhaustiveness at the same time. Besides that, they found that the individual features allowed them to effectively address the solved cases by determining various configurations that represent well-performing solutions, applicable to the requirements of the cases they had to solve (Q14). Lastly, also the efficiency of the approach in the context of the presented problem was stated to be satisfactory, including the time needed for preparing, generating, evaluating and selecting the configurations (Q19).

6.2.3 *Results of evaluating the session realism*

The result of evaluating the realism of the cases solved during the first session was that most of the data was considered available also in the real situation. The participants stated that the input data describing the equipment components, calculation constants (such as worker wages) and demand scenarios were available (Q37). Yet, some product-related input parameters could be specified only with limited confidence, most importantly the equipment handling times and data linked to the physical dimensions of the products (potentially affecting transport and storage cost) and cells (potentially affecting processing times). These factors can have an effect on the accuracy of the results of the performance analysis. Concerning the realism of the cases, the participants stated that all requirements for design and performances of the system configurations were realistic (Q38). While approving the realism of the first case (5 products), the size of the second case (20 products) was found to be a configuration problem too large to be considered in reality.

6.3 DISCUSSION

In the following sections, all aspects will be discussed in a similar sequence as the presented results, relating to the key features of the DSET (Section 6.3.1), their complementary effects (Section 6.3.2) and the session realism (Section 6.3.3).

6.3.1 *Key features of the application*

Considering the variety of configurations (Q1a, b) one participant noticed that it was difficult to assess the plausibility of the product allocation. Moreover, it appeared difficult to compare the system configurations with regard to their allocation (Q4). These findings point towards a problem that was observed already in previous implementations of design automation tools: Chakrabarti and Bligh presented a computational tool that proposes various configurations of a mechanical arm support. The participants of their evaluation experiments found it difficult to interpret the representation of the design solutions in the

support tool [121]. Similar to that, the textual description of production system configurations in the DSET appears to convey too little detail for the production engineers to allow them to interpret the implications and easily identify the differences between solutions. Hence, the solutions' level of abstraction and representation inside the tool appear to be important aspects for further improvement of the DSET. Yet, the participants relativized that manually designing the detailed layout of the chosen concept would represent a means to allow for better interpretation and comparison. Since this is an activity required in any design project with the DSET-based approach, the relevance of the mentioned lack of interpretability has to be seen in relation, as the solutions' feasibility is implicitly verified in the following steps. Additionally, the details of the synthesis routine could be explained to the engineers more elaborately and might improve their understanding of the synthesis procedures and lead to an increased confidence into the plausibility of solutions. Due to the little time available for conducting the session, an extensive explanation was not possible.

An interesting statement made in the course of the verification session was that the participants were not aware of their ability to indirectly influence the system configurations' properties by using requirements (Q2a, b). After receiving the detailed explanation of how to use the requirements to guide the generation of system configurations in the second session, for instance by prescribing the scaling factors, the participants stated that this implicit principle had not been sufficiently clear to them during the first session. It can be assumed that a more thorough explanation of the rationale and purpose of the tool's features will be beneficial to allow the users derive maximum use from the DSET. An indication for this can be seen in the fact that the requirements were acknowledged as means to increase the feasibility of system configurations after elaborating on this aspect. Further opportunities for increasing the users' awareness of the DSET's characteristics are to showcase the features' effects based on multiple example cases and allowing the prospective users to gather more experience with the unusual approach.

When evaluating the users' ability to judge the sensitivity of system configurations by considering the scenario-dependent performances (Q3), the participants expressed their interest to compare the detailed, dynamic performance of the configurations in the course of the years. Even though these year-related performances are calculated during the analysis and therefore could be displayed in theory, the evaluated version of the DSET only provided user interfaces to visualize numerical and textual properties of individual system configurations in the scatterplots and tables. Hence, more suitable formats for visualizing time-series data such as line plots should be integrated.

Another interesting remark was made by one participant when evaluating the ability to identify system configurations that appear promising (Q5): he stated that before being able to evaluate if the set of solutions contained promising and useful candidates, he would need to develop a gut feeling for the problem. This appears an interesting remark since – as already stated in Section 5.4 – familiarity with the problem is probably not present at the outset of the design process. Also, users might require some more practice with the DSET so that a learning curve can be developed. Yet, as a means to create familiarity without requiring the users to manually design and analyze candidate system configurations, the heuristics typically applied by the engineers could be implemented in the DSET and their respective standard configuration types could be synthesized computationally as a benchmark to assess the intrinsic value of the additional configurations.

6.3.2 Complementary effects of the features

When discussing the ability to compare the system configurations with regard to the effectiveness of the design process (Q11), the participants stated the accuracy of the performance analysis would be critical to the usefulness and effectiveness of the comparison:

"[...] the feasibility in itself depends also on the calculation. So, how accurate is the calculation itself."

This highlights the importance of suitable analysis rules and an adequate level of detail of the models in the DSET, both aligned to the users' requirements in the context of the design problem. In the absence of a detailed layout, the analysis rules used for calculating the cycle-times of the products should yield realistic approximations of the real cycle times to generate reliable analysis results. Despite the indicated usefulness of the automated analysis, verifying the suitability of the implemented analysis methods and the related model properties appear vital next steps.

In addition, the participants considered the number of properties for describing each configuration too high, making the DSET confusing at times (Q11). This confusion could lead to an ineffective and inefficient application of the DSET. They stated that some of the analyzed and visualized technical properties were not required to choose configurations if their implications were quantified and covered by other properties. The number of cells of the configurations was specifically mentioned as an example for a performance that could be omitted if the consequences of a higher number of cells – which is mostly higher costs for additional employees that operate these cells – were considered in the cost-performances. Yet, it appears counterintuitive to exclusively implement indicators that are based on the situation-specific goals of the decision-makers, as the properties perceived as not applicable by the users might become relevant in the future. Hence, providing a broad range of properties could be decisive for facing future challenges with the tool in an efficient way and without requiring substantial software modifications in the context of evolving problems.

During the second session the participants suggested an association between factors that had not been covered by the questions asked in the first meeting. They acknowledged the importance of an exhaustive coverage of the prevailing design options (subject in Q20) as an additional influence factor for effectively applying the DSET (Q14). The participants expected that additional design procedures would lead to an even greater variety of the system configurations proposed by the DSET, allowing to find a higher number of suitable solutions. This means that the design-rule sets only covered a share of the possible system configuration types. Hence, implementing additional heuristics could increase the number of suitable candidates for specific objectives and making the approach more effective.

Another remark made by the participants was that it was difficult to know if the presented set contained all feasible design solutions for the specified problem, despite various iterations of adding requirements and generating new system configurations. As described in Section 5.5, the chance to find suitable solutions could be increased by setting up the synthesis runs in the fashion of factorial experiments, where each run represents a unique combination of problem data and requirement values. In this manner, a more strategic approximation of the problem boundaries can be attempted and increase the users' confidence. Nevertheless, as shown in Section 5.2, it can be difficult to state if the system configurations represent the all possible options, which is typical for the use of heuristics [122]. Also, the DSET-based approach embodies a rationale that prioritizes constraint sat-

isfaction over analytic optimality. Receiving a brief introduction to the DSET and solving two example problems in a few hours might not have been enough time for the engineers to get used to this rationale. Hence, giving the users opportunities to internalize the new approach over a longer timespan could improve the understanding of the DSET's rationale and lead to different remarks. In contrast to that, technical solutions to address this remark could be the techniques described in Section 5.2.3.

An interesting remark was made by one participant when asked for the efficiency of the sensitivity analysis of the configuration (Q12):

"I think in most cases it's not done at this moment, just because it takes a lot of time. If you can do it in a click of a mouse, it's quite efficient."

The remark related to the current practice of the company, where sensitivity analyses of system designs were not performed. Using the DSET however was described an efficient approach to conduct sensitivity analyses of the design concepts, hence allowing more thorough assessments of system concepts than practiced in their established design procedures. Another critical aspect with regard to the sensitivity analysis was identified in the sources providing the input for it. For instance, the participants mentioned that the demand scenarios should be provided by their colleagues from the sales department to obtain the highest quality of information. The underlying motivation was stated as follows:

"Basically I think the specification is the phase where the others are most depending on."

The statement points out that preparing the input data and defining the requirements are highly critical activities, that have the highest impact on the system configurations resulting from the synthesis process. Both the source of the information – particularly the product demand – and the way of formulating it in the tool were recognized as important by the participants. In this way, the conclusion from the internal evaluation in Section 5.3 that showed the importance of consciously specifying problem information was independently confirmed.

When presented the main features of the approach – the automated proposal of configurations, comparison, exclusion of infeasible candidates, evaluation of the sensitivity and recognition of promising candidates – and asked for their relation to the effectiveness (Q14), one participant stated the following:

"For me it's basically a roadmap [...]. You have one objective, that's the one stated above (the effectiveness; authors note) and the five factors are the steps you take to reach that objective"

Hence, the collective effects of the individual features were recognized as methodical approach, extending the set of potential system configurations in the beginning, and then gradually comparing and converging towards specific areas of the solution space before making the final selection. At the same time, the participants estimated the DSET would enable them to save time during the phases of preparation and concept generation and confirmed the efficiency of the approach.

"This session alone we just evaluated a hundred configurations, I do not remember when was the last time I looked at so many solutions. [...] It makes it easier to see multiple configurations within short time, and to be able to compare things you need multiple options. And now (relating to the

traditional way of working; author's note) you are normally working so much on one option or two options that you don't have too much time to make the comparison and now (with the tool; author's note) you can dive more deeply into the performance points of view."

6.3.3 Data availability and session realism

When evaluating the realism of the experimental session, one important aspect mentioned by the participants was their reduced ability to specify machine-related input data, specifically the transport times inside the cells. Commonly, the production engineer chooses an explicit set of technologies to execute a production process. That is, for instance, a gripper for holding the product, which is moved by the 7-axis robot towards a stationary machine tool (option 1). In the same way, the robot could also move the machine tool to the stationary fixture holding the product (option 2). While both concepts deliver the same result – a completed production process – the chosen approach may have implications when considering also the preceding and following steps in the sequence of production. If option 2 is chosen and another machine tool is required after completion of the production step, the robot needs to change the tools and requires more time when compared to option 1, where the product could be simply moved to the next stationary machine tool. This means that in the company's established design process, the time required for the entire process chain was treated as the result of the fully specified design concept and not as an input. In this perspective, the difficulty of predicting the processing and movement times inside the cell perceived by the participant appears understandable. Nevertheless, the standardized reconfigurable architecture of the cells in the new concept implies to standardize the used technologies. Creating analysis models of these standards and implementing them in the DSET could help to create a more accurate representation of these options. Also the transport cost rate was mentioned as difficult to specify as an input of the DSET. To address this, a better representation of this property could use the available physical information about the products and their components, enabling to achieve a more exact estimation of the transport cost rate and the influenced logistics cost.

Furthermore, the participants perceived the size of the second evaluation case as unrealistically large, since they would not reconfigure for 20 products simultaneously. At the moment when this statement was made, the company had little relevant experience with the new shop-floor strategy and the implicit system configuration opportunities. Their traditional engineering process was the only reference for "realistic" problem sizes, which however simplified the problem by reducing the number of products beforehand and only considering two options: designing individual, dedicated cells for new products; or allocating additional products to existing universal cells by splitting up all production steps. Thus, the many products of the existing portfolio and hardware of almost 100 existing production cells were not considered again for the design of additional subsystems. Moreover, the company did neither have the required capacity in the design department to design larger systems than traditionally, nor did they have procedures or tools in support thereof except for CAD models and spreadsheet calculations. Hence, the second case was deemed unrealistic at first because it did not match the pattern of problems considered in the established design processes. Yet, it embodied a new size-class for problems that actually can be tackled in the future, enabled by support tools such as the DSET. This explanation for their initial disproving became apparent and was confirmed by the participants in a request to elaborate on the statement.

6.4 CONCLUSIONS

The practical evaluation of the DSET showed the perception of the industrial system designers and allowed to evaluate the applicability and usefulness in context of their design problem, providing an answer to RQ4 (*How applicable and useful is the suggested approach to design automation in practice?*). The participants managed to find system designs matching the various specifications for all experimental cases. The participants confirmed the main features of the DSET and recognized the complementary effects of these features. They found the application of the DSET an effective design strategy for expanding the scope of considered system concepts and consequently converging by comparing the system configurations, imposing requirements and eventually selecting the most promising concepts. The assessment of the configurations' sensitivity gave them additional insight and represents a more extensive evaluation than currently practiced. The participants acknowledged that the preparation and generation phases were vital for the outcome of the DSET-based process, simultaneously offering the biggest potential for time-savings during the development process when compared to their established process. Furthermore it was acknowledged that the approach allows to evaluate more options than in the established design procedure and consider various requirements. A potential domain for improvement identified by the participants was the ability to interpret the feasibility of individual system configurations and – closely related – the textual representation of the system configurations. Also the number of parameters that could be manipulated was perceived as too large. The participants were initially not aware of some functionalities of the DSET, which might be due to the short time available for introducing the DSET during the session and the lack of familiarity with the unusual approach to system design. Hence, the users should get the opportunity to become more familiar with the approach at the outset to get a clearer notion of its implications. The results of evaluating the session realism suggest that the DSET is close to the practical needs of the company. The major part of information used as input for the evaluation problems was available in practice. This suggests that minor adjustments to the model and GUIs can make the DSET applicable to the company's routine processes and puts forward the notion that the efficiency and effectiveness-related benefits can be realized in practice.

VALORIZATION OF THE DSET

This chapter aims to explore further opportunities for valorizing the approach. Section 7.1 therefore describes the estimated effects of the DSET when applied in the design process of the company that evaluated the tool. After that, Section 7.2 describes opportunities for using the approach in combination with other decision-support tools. Section 7.3 identifies potential additions and further fields of application in manufacturing environments, before the conclusions are drawn in Section 7.4.

7.1 ESTIMATING THE DSET'S EFFECTS

Even though the viability of adopting the approach was only qualitatively investigated in Chapter 6, the findings allow to reason about the quantitative implications of the DSET. Using the DSET to design production system configurations differs from the established design procedure in the use-case company regarding the sequence and extent of the individual phases from data preparation until fully specified design. This section therefore presents an estimation of the DSET's effects, comparing the established system design process and the DSET-based approach.

The use-case company's established production system design process starts with the preparation and analysis of the problem data, consisting of the product's expected demand and technical specifications (cf. Figure 63). After getting acquainted with the problem, the production engineers typically choose either to go for a dedicated production cell – a specialized cell for the considered product(s) – or producing the products on the universal cells, where each required type of production process is executed on an individual cell. If the prospective customers have no specific requirements for the production of the part, the decision for either concept is based on the expected production volumes. In case the universal concept is chosen, the feasibility and resulting performance of each production step on the suitable cells is assessed. For developing dedicated concepts, the production engineers determine the needed operations and production equipment required to build cells with the matching capabilities. After specifying the design, they assess if the designed cell performs well enough with respect to the cycle-times of each production step and – if necessary – adjust the design iteratively (see iteration loop in Figure 63). According to the experienced production system designer, the typical duration of the entire process from start to the final concept is around 40 hours.

The design process based on the DSET also starts with the preparation of data. The next step, however, differs in that the users iteratively generate and choose among the many alternatives proposed by the tool (iteration loop 1). After that, the engineer manually develops a detailed design of the system by integrating the building blocks of the production cells into the standardized spatial layout. In the next step, the performances used for selecting the concept are verified by analyzing the detailed layout of the system configurations. In case the performance deviates, detailed layouts could be generated based on one of

the other system configurations previously selected (iteration loop 2); or another system configuration chosen for generating detailed layouts (iteration loop 3).

Figure 63: Comparison of the established design process and the DSET

By comparing the two processes, the time required for applying the DSET can be estimated. The DSET uses digital descriptions of the products, the current configuration and the components that can be used in the configuration. As this information is also available in the current process it can be assumed that the data preparation requires similar time. The calibration of the synthesis runs and their execution can be performed rather quickly (approx. two minutes per run). Once the users have gathered sufficient experience using the tool's features for navigating the design space, fifteen minutes for exploring a single set appears to be a realistic value based on the experience of the author. Hence, generating and exploring four sets of each 100 solutions allows to evaluate a total of 400 solutions in slightly more than an hour's time with the DSET-based approach. After selecting one of the concepts, the duration of the detailed design phase will likely depend on the number of cells and components to be integrated. Yet, it should be comparable in dimension to the previous detailed design, as the components are already specified in the chosen configuration and only need to be arranged into the layout. During the following analysis, the cycle-times need to be determined considering the additional cell details, such as positioning of the FAGs and resulting handling operations, to determine the actual transport and processing times inside the cells. Also for this step, a similar duration as in the current process is assumed. At the end, these times need to be verified by comparing the performances of the detailed design and the rough solution concept to ensure that they do not deviate too much, which should not take longer than one hour. Based on these assumptions and without taking into account the iterations IL2 and IL3, the DSET-based process should not be significantly longer than the current process, however, the number of concepts considered in its course is considerably higher than currently.

When estimating the impact of iterations on the duration of the DSET-based approach, it is important to note that the opportunity to reconfigure the systems requires to standardize the production equipment to some level. For instance, the 7-axis robots that automatically execute the production processes inside the cells need to have the same load specifications so that heavy parts can be produced on all cells and full reconfigurability of the system is achieved. Such standardizations could improve the ability to model the performances of system configurations, which in turn can yield more accurate performance analysis results

at an early moment in the design process. Hence, the higher the match between the analysis results of the DSET and the ones of the detailed designs, the lower the probability of requiring iterations in the later phases caused by performance deviations. The expected first-time-right yield of the DSET-based approach considering the entire design process until the verified system concept therefore allows to estimate the duration of the DSET-based approach similar to the established process. Simultaneously, many more configurations can be assessed, also considering the various problem statements of the users. Hence, choosing the best performing system configurations from the broad range of generated solutions is expected to yield a more effective process than could be achieved manually and also a more efficient approach.

7.2 EMBEDDING THE DSET IN A WORKFLOW

As stated initially, the research described in this thesis was conducted in the frame of the Robust PlaNet project. In the context of the problem considered in the use-case, not only the allocation of the resources and products to multiple cells has to be determined, but also the cells' layout and production plans. To provide support for all problem aspects, one could imagine to develop one model unifying all aspects or integrate various models that are focused on the specific decision aspects. To realize the latter, a support framework was developed in a collaborative effort of the participating research institutes (first described in [123] and extended in [124]). The framework consists of four software tools: the Assembly system configuration tool, the Assembly cell configuration tool, the Production planning and simulation tool and the Reconfiguration planning tool. As the papers describe the approach in detail, this section only provides a brief summary of the main rationale. The constituent tools and the workflow will be outlined in the following, except for the Assembly System Configuration Tool since it refers to the DSET described in Chapter 4.

7.2.1 *Assembly cell configuration tool*

This software was developed in the group of Technology and Production Systems at the Department of Mechanical Engineering of the Politecnico di Milano, representing the result of the work of Massimo Manzini, Marcello Urgo, Marcello Colledani, Andrea Ratti and Alessio Angius. The tool is an application programmed in C++ and Java that can suggest various layouts of a production cell. The starting point for the tool is a set of products and their production steps. Additionally, the equipment required for performing these production processes in alternative ways is used as an input. The Assembly cell configuration tool then generates various layouts, by choosing an equipment set that can execute the production processes and positioning the equipment components inside the production cell. Afterwards, the detailed sequence of operations (such as robot movements and tool changes) is determined and the resulting dynamic performance of the cell analyzed. The output of the tool is a set of different layout alternatives for each cell.

7.2.2 *Production planning and simulation tool*

The Production planning and simulation tool was developed at the Research Laboratory on Engineering and Management Intelligence (EMI) of the Institute for Computer Science and Control (SZTAKI) at the Hungarian Academy of Sciences (MTA) as a result of the research

of Dávid Gyulai, Botond Kádár and László Monostori. The input for this software tool is the detailed layout of the production cells and the durations of the operations taking place on the various production resources inside the cells. Additionally, the order forecast is taken into account. In the first step, a cost-optimal production plan is determined based on the contractual delivery volumes. In the second step, the performance of the multi-cell system configuration and the determined production plan is analyzed by experiments with a discrete-event simulation model. The procedures executing these task were implemented in the FICO Xpress optimization software suite (production planning) and the Siemens Tecnomatix Plant Simulation software (simulation model).

7.2.3 *Reconfiguration planning tool*

The reconfiguration planning tool is associated to the aforementioned Assembly cell configuration tool and was developed by the same group. The tool was implemented in C++ and has the objective to help to determine the reconfiguration strategy for the production cells over a longer period of time. As input, the layouts of the production cells and the optimal production plans are considered. While the layouts for each cell were generated for individual time periods, the reconfiguration tool considers multiple time periods and determines the minimum-cost reconfiguration plan for each cell over the forecast horizon, by selecting the layouts in support of this objective.

7.2.4 *Complementary tool workflow*

The workflow combines the tools and their complementary scopes, as presented in Figure 64: the DSET represents the first tool to be used, resulting in a set of suitable system configurations consisting of one or multiple cells. One of these configurations is then selected by the user to be further detailed in the Assembly cell configuration tool. For each cell configuration, the tool suggests feasible layouts and calculates the detailed task duration and the performances. Afterwards, the procedure evaluates if the layout allows for cycle-times that match the ones used as input for the DSET. In case the times strongly deviate from each other or no feasible layout can be generated, the user can either refine the input data for the DSET and generate new configurations (indicated by feedback loop 1) or try another multi-cell configuration (FL2). In case layouts can be obtained for all cells, the Production planning and simulation tool can be used to determine the minimal-cost schedule for the cells, which is afterwards simulated in the discrete-event model to extend the analysis with stochastic parameters and random events. In case the batch sizes required for generating a feasible production plan are different from the ones used in the Assembly cell configuration tool, they can be updated and the cell layouts verified with the new values (FL 3). If the batch sizes are in accordance with the value previously used in the Assembly cell configuration tool, the Reconfiguration planning tool is used to determine reconfiguration strategies for each cell with the objective of minimizing the lifecycle cost for the expected demand situations.

As a result, the decision-makers should be able to select a specific system concept with the DSET and then determine the system's layout and a plan, which determines the form of reconfigurations in the respective time periods, resulting from the layout generation and optimization tools. The operative control of the production system is supported by the Production planning and simulation tool.

Figure 64: Workflow integrating the decision-support tools

The basic motive of the workflow is to provide a comprehensive and integrated support approach to the design, layout and planning of the production system. It aims at concurrently determining system configurations and operation policies that perform robustly even in the face of an uncertain future customer demand by anticipating this uncertainty at two instances: (i) when initially applying the DSET and (ii) when using the Reconfiguration planning tool. Moreover, the feedback opportunities enable to revisit decisions and refine information used in the previous steps, allowing the users to integrate their requirements and customize the solutions by iterating through the workflow. To achieve this functionality, the models implemented in the various software tools were tailored to the reconfigurable system architecture and harmonized afterwards. The integration of the models was reached by connecting the tools through a web-based platform that also represented the Robust PlaNet project's main outcome and demonstrator, the Simulation and Navigation Cockpit. This web-based platform allows to apply the individual decision-support tools and acts as backbone of the workflow by enabling to exchange the results of the individual tools. While the workflow was only demonstrated but not operationally evaluated in practice, the concept shows how the combination of models with various scopes and at varying levels of details can represent a possible means to manage the considered, interrelated subproblems.

7.3 EXTENDING THE DSET'S CAPABILITIES

While the configuration types in the DSET and the related design-rule sets were developed to suit the company in the project use-case, the evaluations in Chapters 5 and 6 showed that some features of the DSET can be improved. Moreover, the benefits indicated by the evaluation might be transferred to other production-related problems by adding model details, system types and tool features tailored to the considered design problems. Specific suggestions for further development of the DSET that appear promising to the author are briefly sketched out in this section.

ALGORITHMIC PROCEDURES In response to the practical evaluation in Chapter 6, where the engineers stated that even more types of system configurations could have been implemented and appeared worth considering in the tool, further design procedures should be implemented to allow for generating all relevant configuration types. This could make it possible to come up with better performing configurations and a higher number of feasible solutions for specific problems. Furthermore, the engineers expressed their interest in having procedures that can perform partial synthesis of the production system configurations: some parts of the production system could be manually prescribed by the engineers and the

algorithms used to suggest complementary parts of the system. This functionality could be applied when the users intend to keep part of the system configuration, combining a simulation of the fixed part of the system with the computational synthesis of alternatives for the remaining part, allowing to evaluate the performance of the resulting systems more holistically.

MODEL COMPOSITION The model implemented in the DSET could be extended by means of more detailed machine information, such as failure and repair rates for the machines, but also for the considered logistics and storage systems. Similarly, physical details of the products and machines could be added, including information about the spatial configuration of components describing layout of the factory and cells, which would allow to perform more accurate calculations of the performances. Also, the analysis of the configurations could be extended by employing non-deterministic analysis approaches to increase the degree of uncertainty reflected in the model.

TOOL FUNCTIONALITY As explained in Section 5.4, the synthesis and selection strategies play an important role during the process. Software features supporting in these activities might increase the benefits of the DSET. For instance, features that automatically set combinations of the scaling factors, problem specifications or requirements could further reduce the time needed for preparing and generating the synthesis runs. Also, the generation of scenarios could be facilitated by means of an automated parametrization engine that randomizes the values according to the specifications of the users. Similarly, further features that support the users in customized selection (e.g. 3D scatterplots) could add to the efficiency of the approach. Adding such functionalities to the DSET could further improve the effectiveness and efficiency. Also additional GUIs should be considered as means to improve the DSET. Specifically, the lacking ability to evaluate the feasibility of the configurations based on their representation in the support tool (Section 6.3) could be addressed by presenting the detailed features of system designs. This could be structured according to the four system architecture views – physical, functional, quantification and operational – identified by Haveman as essential for communicating simulation results [125]. Concretely, the visualization of available data could be improved by presenting 2D- or 3D-layouts instead of textual descriptions of the system designs; or using line plots to present detailed and dynamic performances of solutions. In addition to the dots in the scatterplots and tables used to present aggregated information, such additional visualization could enhance the ability of the engineers to compare various concepts with respect to their features. Lastly, further effectiveness-related benefits might be realized by enabling users to formulate requirements for the design and performance of individual subsystems, since in the DSET's current form they can be effectively formulated for the numerical (e.g. number of cells, performances) and textual parameters (e.g. design-rule set to be applied) only on the highest level of aggregation.

PROBLEM FOCUS While the DSET was developed to support holistic RMS design, the approach to problem solving is transferable also to other production system design problems. A prototype applying the principles of the DSET to supply chain configuration design – representing yet another class of production system design problems – was successfully developed at the author's department [126]. Moreover, Section 3.2 showed that existing support approaches for automating the design of production system types such as flexible manufacturing systems or conventional assembly lines imply to formulate the design prob-

lem in ways that are similar to configuration problems. The applied solution procedures often presuppose defined solution topologies, requirements and objectives of the resulting systems. This suggests that the DSET can serve as template for further applicable and beneficial tools that support the design of additional production system types.

7.4 CONCLUSIONS

The previous sections argue that the DSET-based approach to designing systems can lead to a higher efficiency of the design process compared to the manual design practice of the company, measured in the effort required per assessed candidate solution. This efficiency eventually enables the users to design more effectively by considering various alternatives and requirements in large design spaces. An opportunity to use the DSET in an extended problem definition with additional design and operational characteristics is shown by the support tool framework that integrates the DSET into a workflow with other support tools. The overall design and management problem is divided into smaller problems that are solved separately, using the integrated, individual tools to determine feasible and compatible solutions of the subproblems. Eventually, suggestions are made for improving the DSET and considering further production system design problems. These suggestions represent opportunities to make the approach applicable to additional production system configuration problems. In this way, the chapter shows how the DSET as support technology might help to utilize the flexibility of manufacturing technology and help to realize its benefits in industrial practice.

SUMMARY, DISCUSSION AND CONCLUSION

This research explored requirements for support tools for the design of reconfigurable manufacturing systems and the application of a novel design automation approach. This chapter summarizes the findings of the previous chapters in the light of the initial research questions and objectives. Furthermore, the contributions and implications are explained and the limitations discussed. After formulating directions for future research, the conclusions are drawn and the research is reflected upon.

8.1 FINDINGS OF THE RESEARCH

After revisiting the results of the evaluation, the research questions as well as the scientific and practical research objectives are recalled and answered respectively.

8.1.1 *Accomplishment of the support tool objectives*

The evaluations in Chapters 5 and 6 showed that the DSET's features allowed to achieve their respective direct objectives, as the tool could be efficiently used in various RMS design problem formulations and was capable of generating various and applicable configurations for each case. Moreover, the qualitative evaluation has shown that the prototype was considered useful and mostly applicable to the company's design practice. Nevertheless, the achievement of the superordinate objectives of the support tool – an efficient and effective system configuration design process – has to be seen in relation to its application environment and is discussed in the following.

EFFECTIVENESS The internal and practical evaluations confirmed that the DSET allows for an effective approach to RMS design, since it enabled the users to consider subsystem design decisions holistically, identify relevant properties (e.g. cost-drivers in complex formulations of the design problem) and their dependencies. The DSET thus supports the selection of suitable system designs in spite of multivariate and complex problem specifications. Yet, the results also highlighted two factors that require a relative view of the claim of effectiveness. The system designers interviewed in the practical evaluation stated that the variety of solutions that can be obtained with the implemented design-rule sets is considerable, yet further configuration types and associated design spaces are imaginable and not covered by the implemented design-rule sets (see Figure 65). As proposed in Section 7.3, covering these additional solution types by including additional design procedures could allow to further increase its effectiveness. Similarly, the theoretical evaluation in Chapter 5 showed that the design synthesis procedure helps to achieve various solutions. However, the semi-random instantiation of design parameters can imply that only a fraction of the potential solutions for the problem specifications are generated (see box in Figure 65).

Therefore, the effectiveness of the DSET can be considered satisfactory but can be further enhanced.

Figure 65: Effectiveness-related aspects at a glance

EFFICIENCY The tool’s efficiency also has to be seen in relation to the efforts required for its development and application. When applying the tool in any of the described experiments, it took only few seconds to generate the entire set of solutions on a Dell Latitude E6540 laptop PC (CPU: Intel Core i7-4800MQ @ 2,7GHz; memory: 8GB RAM). Since it took less than four hours to identify suitable system designs for two realistic sample problems during the practical evaluation, the tool’s efficiency of application can be considered adequate. An important feature in this context is that the automated and requirement-based filtering of unsuitable solutions allows the users to focus on assessing the relevant solutions only. Another aspect related to the DSET’s efficiency is the opportunity to use it to assess the solutions of related problem statements. The DSET does not require modifications of the knowledge base even if the problem sizes (e.g. number of products and processes) and their formulation change (e.g. moving from cost requirements to design requirements) as long as the basic representation of the problems – i.e. component properties, parameters of interest, system configuration types and properties – is the same. Hence, by changing the input data that describes the machine instances, products and demand scenarios, as well as the requirements and objectives for solutions, the tool can be used to assess multiple problem statements and can be applied in multiple reconfigurations, allowing both the distribution and justification of the development efforts. Developing the algorithms for the specific RMS architecture in this research took an equivalent of approximately 30 weeks full-time work. This does not include the effort for knowledge elicitation since it was mainly conducted by the Master’s students and was therefore difficult to quantify. This figure also does not include the development of the visualization and interaction components of the DSET because they are not exclusively dedicated to the considered design problem and can be reused in the context of additional problems. The 30 weeks have to be compared to the expected improvements with regard to system designs and the productivity of the design process. Whereas the latter is estimated in Section 7.1 and can be considered significant, the scale of savings required to offset the DSET’s development efforts equals the value of a single robotic tool (e.g. clinching gun) in a configuration of the considered system architecture. Since this is only a small fraction of the total investment of the reconfigurable cells, it appears possible to justify both the development of the model and algorithms and the application of the DSET economically.

8.1.2 Answers to the research questions

The following paragraphs outline the gained insights with regard to the initial research questions.

RQ1: What are the requirements for design automation tools to support the configuration design of reconfigurable production systems? The challenges and issues of designing general and reconfigurable production systems and resulting requirements for support tools for design automation were described and discussed in Chapters 2 and 3 respectively. RMS design is characterized by a large number of possible and distinct technical solutions. It is different from conventional systems, because the system's architecture needs to be defined in more detail upfront to make the system reconfigurable. This makes computational tools that involve an automation of the design task most suited for the configuration design phase, since sufficient knowledge about the system's structure and associated design spaces is only available at this stage. Nevertheless, the design spaces are shaped by the requirements of the decision-makers and it is difficult for them to anticipate whether system designs exist that satisfy their combinations of requirements. Therefore, an important feature of computational support tools is that the decision-makers can test various problem statements and determine their effects on the resulting design spaces. In this context, computational support tools have to be capable of generating solutions that are applicable to the requirements, which calls for algorithms that cover vast design spaces to provide effective support. When multiple solutions satisfy the decision-makers requirements, a support tool should include features that allow the users to assess the quality of the system designs, i.e. for navigating the design space, comparing the system designs and selecting the most suitable ones. To perform this assessment, all relevant properties of the production system configurations need to be assessable by the decision-makers to allow a holistic evaluation of the technical and economic implications and to make justifiable design decisions. To efficiently apply the support approaches, the speed of performing all involved steps – tool calibration, design synthesis, analysis and evaluation – is a relevant factor. Equally important is to provide customizable support tools that can be used to generate solutions for related problem statements and in multiple reconfigurations. This enables the repeated application of the tools and is another determinant of the efficiency of the support approach.

RQ2: How can the design of reconfigurable production systems be supported by means of automated design synthesis? The DSET introduced in Chapter 4 represents a novel approach to design automation, designed to address the previously identified objectives in the context of the Robust PlaNet project use-case. It automates the execution of three heuristic design-rule sets for design synthesis, as well as the procedures for analysis and requirement-based evaluation. As a result, it can quickly generate a high number of production system configurations matching the decision-makers specifications. Representing a compromise between the model's applicability to various problems and its efficient application, the synthesis procedure determines system design parameters in a semi-random fashion and can be guided by applying scaling factors for the synthesis and requirements for filtering solutions based on their characteristics. The DSET's GUIs allow for exploring the generated sets of system configurations, comparing them, assessing their sensitivity and identifying superior solutions. Collectively, these factors enable the users to select the final candidate solutions manually based on their objectives. In this way the tool allows the quantification of the implications of uncertain future demand by visualizing the influence of various product

demand scenarios on the performance of the system configurations and enables their holistic comparison by providing information about their design, as well as technical and economic performances. The speed of the synthesis procedure combined with opportunities for customizing input data, requirements and objectives make it possible to efficiently perform multiple iterations with various design problem statements and assess the resulting design spaces.

RQ3: What are the characteristics of the suggested approach to design automation? The detailed assessment of the DSET's implications was performed by analyzing the approach in experimental cases in Chapter 5. The first experiments focused on the exhaustiveness of the semi-random design synthesis procedure. It was found that the implemented strategy for sampling the design space can imply an incomplete record of the design space. Nevertheless and more importantly, it enables the users to obtain various solutions and expand the range of considered system designs considerably compared to manual approaches. In addition the impact of the user specifications with respect to problem data, design and performance requirements was demonstrated. One result of these experiments was that the analysis of capacity requirement performed as part of the synthesis procedures can have a significant impact on the possible variety of solutions. Similarly, the requirements for the resulting solutions can strongly influence the possible number and types of system configurations. By assessing the similarities and differences of the system configurations found for various problem formulations, it was shown that also the synthesis and selection strategies decisively influence the features of the resulting solutions. In this context, it became apparent that the suitability of the solutions suggested by the design heuristics is situation-specific and using various heuristics simultaneously can increase the range of suitable solutions when considering various problem statements. The essence of these findings is that the users should be aware of the problem specifications and potential degrees of freedom for varying the requirements and the formulation of the problem, promoting to take an experimental approach to production system design to identify the most promising system configurations.

RQ4: How applicable and useful is the suggested approach to design automation in practice? The tool's applicability and usefulness in practice was examined in Chapter 6. For this purpose, a qualitative evaluation was conducted with experts from the company, whose RMS design problem was used as pattern for developing the model and algorithms. The evaluation showed that the experts could successfully apply the tool to assess and select various design solutions for the problems presented to them. The participants perceived the tool as useful in their practice, even though the rationale of the approach was new for them. The results indicate that a more thorough explanation of the tool's features, training or practical application by the experts might be needed to fully utilize the capabilities of the tool, which should be taken into account when evaluating the significance of the findings. Yet, the participants appraised the DSET as a methodical approach to system design that enables them to significantly and efficiently increase the types and number of configurations considered in the design process. Lastly, the DSET was shown to be likely applicable in the company's design practice.

8.1.3 Accomplishment of the research objectives

The starting point of the presented research were the practical and scientific objectives:

The practical objective of this research was to support designers in leveraging the flexibility of reconfigurable manufacturing systems and making the production system design process more efficient and effective.

The scientific objective of this research was to examine how automated design synthesis can be applied to support users in designing configurations of reconfigurable manufacturing systems.

Three studies were conducted to reach these objectives: a literature-based descriptive study for determining the objectives of support tools in the context of RMS configuration design; a prescriptive study that describes the technical implementation and intended functionality of a new tool; and an empirical descriptive study for evaluating the tool's characteristics, applicability and usefulness. The practical objective was addressed by developing the support tool, empirically evaluating it to confirm its effects and identifying potential aspects for improvement. The results show that the suggested approach gives the users insight into numerous and various opportunities for configuring the reconfigurable system and provides features that allow its efficient and effective application. The suggestions made for improving the tool and applying it in connection with other support approaches illustrate further opportunities and benefits for design automation in production system design practice. The scientific objective was addressed by analyzing the problem of RMS configuration design and by clarifying the required characteristics of support tools. These requirements show that developing support tools for production system design is a challenging task, as it requires to intersect the life-cycles of products, production systems and the design support tools themselves. In addition, the detailed technical description of the design support tool and its evaluation approach and results are relevant outcomes from the scientific perspective.

8.2 CONTRIBUTIONS AND IMPLICATIONS

This research makes three contributions to the state-of-the art, which are highlighted in this section. The main domains of knowledge in engineering design to which these contributions can be allocated, are visualized in Figure 66.

Figure 66: Allocation of academic contribution (based on [127])

FRAMEWORK FOR COMPUTATIONAL SUPPORT TOOLS FOR RMS DESIGN The framework with the technical requirements for support tools in Section 3.1.3 summarizes the problem-related aspects identified in the literature that should be considered when de-

veloping computational tools to support decision-makers in RMS configuration design by means of automated synthesis. Hence, the framework can serve as a reference and provide orientation for researchers and practitioners intending to evaluate or develop support tools for the problem considered here, as well as other RMS design problems. Not much research can be found on the relation between production system design problems and the resulting requirements for support approaches. Wu presents a general framework for computationally supporting the management of the main production system design phases [101]. While his framework focuses on the relation between design process phases, documentation and data structures, the research does not address particular design problems or problem stages, neither concrete approaches to problem-solving. Hence, the framework developed in this thesis represents a more problem-oriented set of criteria for developing and evaluating support tools and thus represents a relevant extension.

COMPUTATIONAL SUPPORT TOOL The DSET shows how the design of alternatives can be automated while providing opportunities to customize the specifications of problems and solutions. The software tool represents a prototype that enables decision-makers to assess large design spaces by using automated synthesis and was developed based on the framework. The concrete proof-of-principle demonstrates a technical solution for the previously identified requirements and can serve as blueprint or benchmark for scholars and managers looking for ways of implementation. The implicit novelty of the approach is that the tool complements the decision-makers in the problem-solving process, instead of largely excluding them as observed in many of the approaches presented in Section 3.2. This difference can be also described by the form and moments of stating the preference information, as illustrated in Figure 67 in relation to the activities of knowledge modeling, model calibration and design problem solving. In the approaches in Section 3.2, the preference information is mostly coupled with the design model and therefore difficult to change. In contrast to that, the DSET makes it possible to postpone this statement of preference until the initialization of the synthesis runs, the application of requirements or even until the visual evaluation of solutions by the users. This makes it possible to efficiently apply the support tool in various situations. It was recently concluded that identifying the need for reconfiguration of RMS and expressing it are major issues and still unsolved [74]. In this regard, the DSET enables the decision-makers to opportunistically check if and how reconfigurations of the system allow for improvement of the system characteristics.

EVALUATION APPROACH AND RESULTS The approaches and results of evaluating the tool theoretically and practically provide an indication of the DSET's implications, usefulness and applicability. The insights into the capabilities of the support tool perceived by the practitioners enable the readers to evaluate the maturity and suitability of the approach. Furthermore, the identified issues can be useful for improving the approach or adopting some of its useful features when developing future support tools for design automation. In comparison to the way and scope of the evaluations commonly performed for design automation models in production system design, the evaluation of the DSET encompassed a broader range of criteria. Most of the prescriptive approaches presented in Section 3.2 are evaluated by applying the suggested procedures to sample problems. Such evaluation was performed in a similar fashion for the DSET in Chapter 5, whereas the detailed empirical evaluation in Chapter 6 – focusing on the suitability of using the tool in the context of a practical system design process – represents an option that goes beyond the common practice. Performing such evaluation is crucial for assuring the usefulness and identify-

Figure 67: Preference information coupling points (based on [30, 97])

ing potential obstacles, as well as mandatory circumstances for application and therefore should become customary for research into prescriptive design support approaches. As a result, both the design and the results of the evaluation represent important contributions.

8.3 LIMITATIONS

While conducting the presented research, various choices had to be made that impact the significance and generalizability of the findings. Therefore, the following issues have to be considered when interpreting the results.

PROBLEM ASPECTS With regard to the literature review in Sections 2.1 to 2.3, an important issue is that the identified aspects of the production system design problem might not represent all relevant characteristics that play a role when determining the required support tool features. Since the design and performance variables in production system design problems are highly interdependent, additional variables that need to be considered during the design decision can require tool characteristics that are not covered by the framework (e.g. spatial configuration aspects). As the efforts and complexity of addressing the presented factors already appeared considerable, it was considered more useful to prioritize developing a matching support tool rather than establishing further requirements that could complicate this development. Further research should identify underexposed issues, integrate them into the framework and examine how support tools could fulfill them.

SOLUTION APPROACHES Another limitation is that Section 3.2 does not present a catalogue of all existing computational tools that automate the design of production systems. Hence, further approaches might be found in literature that partly or even fully fulfill the support functionality indicated by the framework. Production system design is a domain of interest for many research disciplines (e.g. computer science, mathematics, engineering design science, management sciences) and a large number of computational approaches

to solving production system design problems is reported using discipline-specific terminology which makes it difficult to identify all relevant approaches. For this reason, the discovery of related work could be facilitated by generating an overview of the terminology used in the various disciplines to refer to the problem related to production system design. Another issue often encountered during the review was that differences in the extent of description and evaluation made it difficult to compare the publications with respect to the problem considered, as well as the support tool implementations and implications. Even though a broad range of publications was reviewed, many of them could not be presented in this thesis since the relevant properties were only partly described and, hence, not suited for comparison. To address this challenge, standards for describing the properties of support approaches should be developed that can guide authors who report their work, so that researchers and practitioners can get an easier overview of existing approaches and understand the capabilities of contemporary support methods.

IMPLEMENTATION Another limitation applies with regard to the characteristics of the generated system configurations in both conducted evaluations. The algorithmic procedures for production system synthesis and analysis involved writing more than four thousand lines of code and small errors, such as a minus instead of a plus, can have serious effects. Since faulty procedures may show noticeable effects only in rare cases, it is difficult to universally guarantee the correctness of the procedures' results. The design synthesis and analysis procedures were intensively debugged and a number of system configurations were manually verified to diminish the chance of unintended errors. Yet, before deploying the tool in the real design process, further testing and verification should be performed.

QUALITATIVE EVALUATION IN A SINGLE CASE An important implication of the findings of the qualitative evaluation is that the DSET was only evaluated in a single case study and the findings therefore might not be generalizable. The time required for knowledge elicitation, development of the customized model and the DSET, as well as the preparation and execution of the evaluation was considerable. The development and examination of the effects in additional cases was not possible in the time frame of this research. Hence, further evaluations should be performed to verify the demonstrated effects. It also has to be considered that the DSET only represents a prototype and the company had an interest in continuing the collaborative development. Therefore, the evaluation result might be biased and can imply that the DSET's effects could have been rated more favorably than actually perceived by the interviewees. To prevent biased results, the format of the research sessions was designed purposefully to stimulate the interviewees to critically reflect the DSET's functionality and effects by using direct questions addressing the DSET's features and objectives during the experimental session, as well as performing an entire session only dedicated to validating the findings. To verify the findings, the actual application of the tool in the real design process could be observed and further qualitative research performed in other problem cases.

8.4 FUTURE RESEARCH

In the following section, additional research opportunities are presented.

CUSTOMIZABLE, EXHAUSTIVE AND EFFICIENT DESIGN AUTOMATION PROCEDURES
An important finding of the Descriptive study I is that the developers of design automa-

tion procedures have to find a compromise between the time required for generating sets of solutions, the exhaustiveness of the generation procedure and the customizability of the design problem statement in the support tool. As indicated by the evaluation in Chapter 5, the exhaustiveness of the design synthesis procedure currently implemented in the DSE is somewhat reduced in favor of a more efficient approach to performing iterations and customizable design problem statements. Research into synthesis procedures that are similarly customizable, efficient in application and more exhaustive could allow to further increase the considered range of solutions and thus help to make the design process even more effective.

DESCRIPTIVE RESEARCH WITH REGARD TO RMS DESIGN PRACTICE AND SUPPORT TOOLS The design of RMS has been extensively studied over almost two decades. However, a design theory attempting to comprehensively describe the various activities and lifecycle phases was only suggested recently. Yet, this theory is mainly based on prescriptive literature. Hence, even though this represents an important step forward, a lot more research is required into the practice of RMS design, which was also acknowledged by the authors that synthesized the RMS design method [74]. Moreover, as shown in Appendix A.5, only sparse empirical evidence can be found for the practical application of the many prescriptive models found in production system design literature. To examine which factors prevent the materialization of the benefits of these methods, empirical insights should be developed that discuss the suitability and issues of the various approaches for a broad range of typical system design problem(s) and their characteristics. In this context, an empirical analysis of the DSET's application in industrial practice could be a suitable start, as well as evaluations of existing approaches in practice analogous to the evaluation in Chapter 6.

MULTI-DISCIPLINARY RESEARCH INTO PRODUCTION SYSTEM DESIGN Other directions for future research into production system design include investigations that combine knowledge from multiple disciplines. As stated in the previous section, the design of production systems is a research topic studied already for decades in various disciplines. The research presented in this thesis attempts to improve production system design practice by considering and combining theories and techniques from some of those disciplines. Nevertheless, the topic certainly offers interesting and promising opportunities for further research that attempts to integrate the various disciplines to reach synergetic effects. For developing future support tools, also comparably new research fields should be considered, such as visual analytics. Research in this field attempts to facilitate analytical reasoning processes in the face of complex and dynamic data [128] and thus might offer further beneficial techniques for making design spaces and the implications of system design preferences accessible for the users. After all, developing a common terminology could be an important first step towards enabling such interdisciplinary research.

8.5 CONCLUSIONS

The functional framework for computational tools to support in the RMS configuration design problem embodies a novel approach to defining the required functionality of support tools. The DSET demonstrates an approach for automating the generation of sets of system designs that actively involves the decision-makers into the generation and exploration of design spaces. The essential notion of the tool is to move the scope of decision-making away

from solely considering a few system properties and a single production system design and – instead – look at various related problem statements, the associated solutions and their properties. The approach to evaluating the tool was informed by the Design Research Methodology and represents a template for future research of prescriptive approaches in production system design. The insights gained from the evaluation confirm that the tool's intended effects were achieved. Even though limitations apply to the research, the benefits of the design automation approach are now established and an agenda for further research to actualize these benefits was presented.

After all, this research shows that customizable tools can complement the users in solving design problems. The DSET can help to utilize the capabilities of flexible manufacturing technology and simultaneously represents a valuable type of artifact, which was anticipated in the introduction of a well-known book on engineering design [55].

“In the future, (...) variant designs will be largely undertaken by the computer, leaving designers free to concentrate on new designs (...). These tasks will be supported by computer tools that enhance the creativity, engineering knowledge and experience of designers. The development of knowledge-based systems (expert systems) and electronic component catalogues will increase the ease with which information can be retrieved, including specific design data, details of standard components, information about existing products as well as their design processes and other design knowledge. These systems will also aid the analysis, optimisation and combination of solutions, but they will not replace designers. On the contrary, the decision-making abilities of designers will be even more crucial because of the very large number of solutions it will be possible to generate, and also because of the need to coordinate the inputs from the many specialists now required in modern multidisciplinary projects.”

This quote highlights the relevance of the proposed design automation approach and – propelled by the results of this research – raises the expectation that DSET-like tools will represent integral pillars for designing future factories.

8.6 RESEARCH REFLECTION

As the very last section of this thesis, I would like to give the readers insight into my own perception of the research. The result that might benefit most from extending the research beyond what is described in this thesis is the reference model. The developed reference model was inspired by considerations that system designers might have in practice, however, it was developed based on literature. As the support tool's design and also the evaluation were based on this reference model, it was only empirically tested in a single industrial company. There is thus a chance that it does not reflect all functionalities and effects that are required by such tools in practice. Hence, the validation of the model in further empirical cases and its extension with additional criteria that are relevant for supporting system designers are important activities and should be main focus points of future research.

In contrast to that, I perceive the strongest result of this research the well-explained and evaluated approach to practically support the designers in defining and solving RMS configuration design problems. The detailed explanation of the problem features the tool aims to address and the functionality provided for this purpose make it possible to understand why and how particular tool features were designed in this specific way and to which extent they actually work.

Next to the relative quality of the research results, I retrospectively can also highlight particular phases of the research process that went well and others, which I would approach differently. If it were possible to start over with the research project with my current knowledge, I would aspire to perform the practical evaluation at an earlier point in time and attempt to have multiple iterations between the development of the support tool and observing practitioners when using it. This could give me the chance to improve the support tool more quickly and to such an extent that the evaluation could be performed in the real design process instead of a staged one. The evaluation in the real design process could not only increase the relevance of the detected effects of the tools, but also allow to improve the (range of the) algorithms and the quality of the user interfaces.

Yet, there were also research phases that went well in my perception, most importantly the design of the algorithms and the design and execution of the empirical study in industry. Firstly, the synthesis algorithms had to reflect a totally different mindset than previous approaches. While the design of algorithms that perform a fully automated selection, such as optimization algorithms, is guided by the question *“How can it be made sure that the best possible solution is selected?”*, the tool can be characterized by two questions different from that, namely *“How can it be made sure that the tool can come up with all potentially relevant solutions?”* and *“Which instruments/KPIs can be provided to the decision-makers so that they can exclude the irrelevant solutions?”*. Hence, the design of the algorithms went smoothly, considering that approaches addressing these different questions could not be found in literature and therefore no references for comparison existed. Secondly, the evaluation of the tool by the industrial designers had to reflect these particularities and therefore required to develop a new evaluation format, which additionally had to be conducted in a narrow time window set by the industrial decision-makers and not by the researcher. Despite the limited time, the design of the evaluation allowed statements of confirmatory nature with regard to the objectives of the tool, but also allowed to explore potential points for improvement that might be needed when its practical application is pursued. The focus of this structured assessment reflects the rationale of the tool *“Who else, if not the industrial expert facing the design problem, has the best pre-conditions to evaluate the quality of a design/design support tool?”*.

The results of these research processes indicate significant future potential of the tool with regard to its practical use. As the evaluation showed, the system designers perceived some drawbacks that, however, do not appear to substantially limit the tool's intended effects. That means that even though the approach does not yet exhibit all characteristics desired by the production engineers, it actually could be already employed in practice and indicates directions for future support models that are broadly applicable and economical in use. In addition, the detailed explanation of the background, reference model and life-cycle perspective on support tools represent results of high practical value. They can be used as a guideline when managers intend to estimate the major factors that have to be considered when deciding to buy or develop support tools.

Altogether, the research advanced and partly also changed my basic understanding of production system design and its support by means of computational tools. At the outset of this research, I frequently found a specific argument in publications, stating that design support tools are needed because the design problems were so complex that humans are struggling with solving them. Initially, I thought that this argument was true but now - after completing my research - I changed my opinion. Today, I do not think the reason why engineers (should) use support tools (also in the format of spread-sheet calculations)

for product and system design is that they are incapable of developing well-performing solutions. On the contrary, I believe that true experts cannot only develop good technical solutions, but they also have the creative ability to develop such a quantity of relevant questions and problem derivatives that finding answers and solutions to all of them simply takes too long. That's why I think support tools should ideally not attempt to predefine the questions and answers to the technical challenges but provide functionalities that anticipate the needs of the decision-makers when approaching the broadly defined design problem from various sides.

BIBLIOGRAPHY

- [1] J. F. Kennedy. *Public papers of the presidents of the United States: John F. Kennedy: Containing the Public Messages, Speeches, and Statements of the President, 1961-1963*. Public Papers of the Presidents of the United States: John F. Kennedy: Containing the Public Messages, Speeches, and Statements of the President, 1961-1963. U.S. Government Printing Office, 1963.
- [2] Eurostat. *Sectoral analysis of key indicators in manufacturing (NACE Rev. 2 - Section C)*. Eurostat, 2014.
- [3] CIRP, Luc Laperrire, and Gunther Reinhart. *CIRP Encyclopedia of Production Engineering*. Springer Publishing Company, Incorporated, 2014. ISBN: 3642206166, 9783642206160.
- [4] M. Bellgran; K. Säfsten. *Production development - design and operation of production systems*. Springer-Verlag London, 2010.
- [5] Henning Kagermann; Wolfgang Wahlster; Johannes Helbig. *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0*. Tech. rep. Forschungsunion Wirtschaft - Wissenschaft, 2013.
- [6] James Manykia; Michael Chul; Mehdi Miremadi; Jacques Bughin; Katy George; Paul Willmott; Martin Dewhurst. *A future that works: automation, employment, and productivity*. Tech. rep. McKinsey&Company, 2017.
- [7] Heebyung Koh; Christopher L. Magee; „A functional approach for studying technological progress: application to information technology.” In: *Technological Forecasting and Social Change* (2006).
- [8] Katherine Nölling; Michael Alexander; Wolfgang Bernhart; Carsten Rossbach. *Smart strategies for smart sensors*. Tech. rep. Roland Berger, 2017.
- [9] Maarten Essers; Tom Vaneker. „Design of a decentralized modular architecture for flexible and extensible production systems.” In: *Mechatronics* (2016).
- [10] Sabina Jeschke; Christian Brecher; Houbing Song; Danda B. Rawat, ed. *Industrial Internet of Things - Cybermanufacturing Systems*. Springer International Publishing Switzerland, 2017.
- [11] Maarten Essers. „Design of a novel, hybrid decentralized, distributed, modular architecture for manufacturing systems.” PhD thesis. University of Twente, 2016.
- [12] Laszlo Monostori; Botond Kádár; Thomas Bauernhansl; Shinsuke Kondoh; Soundar Kumara; Gunther Reinhart; Olaf Sauer; Günther Schuh; Wilfried Sihm; Kanji Ueda. „Cyber-physical systems in manufacturing.” In: *CIRP Annals - Manufacturing Technology* (2016).
- [13] Richard Dobbs; James Manyika. *Disruptive technologies: Advances that will transform life, business, and the global economy*. Tech. rep. McKinsey&Company, 2013.

- [14] Stefan Bley; Christoph Kigler; Jochen Vogel. *Industrie 4.0 – das unbekannte Wesen?* Tech. rep. Ernst & Young GmbH, 2016.
- [15] Jim Browne; Didier Dubois; Suresh Sethi; Kathryn Steckel. „Classification of Flexible Manufacturing Systems.” In: *The FMS Magazine* (1984).
- [16] Hoda A. ElMaraghy. „Flexible and reconfigurable manufacturing systems paradigms.” In: *International Journal of Flexible Manufacturing Systems* 17.4 (2005), pp. 261–276. ISSN: 1572-9370. DOI: 10.1007/s10696-006-9028-7.
- [17] Tullio Tolio, ed. *Design of flexible production Systems*. Springer-Verlag Berlin Heidelberg, 2009.
- [18] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel. „Reconfigurable Manufacturing Systems.” In: *Manufacturing Technologies for Machines of the Future: 21st Century Technologies*. Ed. by Anatoli I. Dashchenko. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 627–665. ISBN: 978-3-642-55776-7. DOI: 10.1007/978-3-642-55776-7_19.
- [19] G. Meng, S. S. Heragu, and H. Zijm. „Reconfigurable layout problem.” In: *International Journal of Production Research* 42.22 (2004), pp. 4709–4729.
- [20] Frans Anne José Ruffini. „Production system design - from practice to theory.” PhD thesis. University of Twente, 1999.
- [21] Jessica Bruch. „Management of design information in the production system design process.” PhD thesis. Mälardalen University, 2012.
- [22] Cecil Bozarth and Chris McDermott. „Configurations in manufacturing strategy: a review and directions for future research.” In: *Journal of Operations Management* 16.4 (1998), pp. 427–439. ISSN: 0272-6963. DOI: [https://doi.org/10.1016/S0272-6963\(98\)00022-9](https://doi.org/10.1016/S0272-6963(98)00022-9).
- [23] George Chrystosolouris. *Manufacturing systems - theory and practice*. Springer Science+Business Media Inc., 2006.
- [24] Sigrid Wenzel; Matthias Weiß; Simone Collisi-Böhmer; Hogler Pitsch; Oliver Rose. *Qualitätskriterien für die Simulation in Produktion und Logistik*. Springer-Verlag Berlin Heidelberg, 2008.
- [25] Mohsen Jahangirian, Tillal Eldabi, Aisha Naseer, Lampros K. Stergioulas, and Terry Young. „Simulation in manufacturing and business: A review.” In: *European Journal of Operational Research* 203.1 (2010), pp. 1–13. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2009.06.004>.
- [26] Ashkan Negahban and Jeffrey S. Smith. „Simulation for manufacturing system design and operation: Literature review and analysis.” In: *Journal of Manufacturing Systems* 33.2 (2014), pp. 241–261. ISSN: 0278-6125. DOI: <https://doi.org/10.1016/j.jmsy.2013.12.007>.
- [27] Wayne L. Winston. *Operations research*. Brooks/Cole, 2004.
- [28] El-Ghazali Talbi; Farouk Yalaouil; Lionel Amodeo, ed. *Metaheuristics for Production Systems*. Springer International Publishing Switzerland, 2016.
- [29] El-Ghazali Talbi. „Combining metaheuristics with mathematical programming, constraint programming and machine learning.” In: *Annals of Operations Research* 240.1 (2016), pp. 171–215. ISSN: 1572-9338. DOI: 10.1007/s10479-015-2034-y.

- [30] Wouter O. Schotborgh; Chris McMahon; Fred J.A.M. van Houten. „A knowledge acquisition method to model parametric engineering design processes.” In: *International Journal of Computer Aided Engineering and Technology* (2012).
- [31] Yann Collette; Patrick Siarry. *Multiobjective optimization - principles and case studies*. Springer-Verlag Berlin Heidelberg, 2004.
- [32] Gary M. Stump; Mike Yukish; Timothy W. Simpson; John O'Hara. „Trade space exploration of satellite datasets using a design by shopping paradigm.” In: *Proceedings of the IEEE Aerospace Conference 2004*. 2004.
- [33] Eugen Rigger; Clemens Münzer; Kristina Shea. „Estimating the potential of state of the art design automation - tasks, methods and benefits.” In: *Proceedings of the DESIGN 2016 14th International Design Conference*. 2016.
- [34] Johan Malmqvist. „Towards Computational Design Methods for Conceptual and Parametric Design.” PhD thesis. Chalmers University of Technology, 1993.
- [35] Joel Johansson. „Automated computer systems for manufacturability analyses and tooling design.” PhD thesis. Chalmers University of Technology, 2011.
- [36] Mikael Cederfeldt and Fredrik Elgh. „Design Automation in SMEs – Current State, Potential, Need and Requirements.” In: *Proceedings of the international conference on engineering design ICED 2005*. 2005.
- [37] Amaresh Chakrabarti; Kristina Shea; Robert Stone; Jonathan Cagan; Matthew Campbell; Noe Vargas Hernandez; Kristin L. Wood. „Computer-Based Design Synthesis Research: An Overview.” In: *Journal of Computing and Information Science in Engineering* (2011).
- [38] Juan M. Jauregui-Becker; Guido Tosello; Fred J.A.M. van Houten; Hans N. Hansen. „Performance evaluation of a software engineering tool for automated design of cooling systems in injection moulding.” In: *Proceedings of the 46. CIRP Conference on Manufacturing Systems 2013*. 2013.
- [39] Marcus Vinicius Pereira Pessôa; Luís Gonzaga Trabasso. *The Lean Product Design and Development Journey*. Springer International Publishing AG, 2017.
- [40] Duward K Sobek; Allen C Ward; Jeffrey K Liker. „Toyota's principles of set-based concurrent engineering.” In: *Sloan Management Review* (1999).
- [41] J. Unglert, S. Hoekstra, and J. Jauregui-Becker. „Evaluating a computational support tool for set-based configuration of production systems: Results from an industrial case.” In: *12th IEEE System of Systems Engineering Conference (SoSE)*. 2017, pp. 1–6. DOI: 10.1109/SYSOSE.2017.7994939.
- [42] David Shahan; Carolyn Seepersad. „Implications of Alternative Multilevel Design Methods for Design Process Management.” In: *Concurrent Engineering: Research and Applications* (2010).
- [43] Christoffer Levandowski; Marcel T Michaelis; Hans Johannesson. „Set-based development using an integrated product and manufacturing system platform.” In: *Concurrent Engineering: Research and Applications* (2014).
- [44] Dag Raudberget. „Practical Applications of Set-Based Concurrent Engineering in Industry.” In: *Journal of Mechanical Engineering* (2010).
- [45] Lucienne T.M. Blessing; Amaresh Chakrabarti. *DRM, a Design Research Methodology*. Springer-Verlag London Limited, 2009.

- [46] Juan M Jauregui-Becker; Botond Kádár; Marcello Colledani; Nicole Stricker; Marcello Urgo; Johannes Unglert; Dávid Gyulai; Emanuel Moser. „The RobustPlaNet Project: Towards Shock-Robust Design Of Plants And Their Supply Chain Networks.” In: *IFAC-PapersOnLine* (2016).
- [47] M.P. Groover. „Automation, Production Systems, and Computer-integrated Manufacturing 2nd ed.” In: *Assembly Automation* 22.3 (2002). DOI: 10.1108/aa.2002.22.3.298.2.
- [48] Serope Kalpakjian; Steven R. Schmid; K.S. Vijay Sekar. *Manufacturing engineering and technology*. Pearson Education South Asia Pte. Ltd., 2014.
- [49] Mohamed Kashkoush and Hoda A. ElMaraghy. „Generating Master Assembly Sequence Using Consensus Trees.” In: *Enabling Manufacturing Competitiveness and Economic Sustainability: Proceedings of the 5th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2013), Munich, Germany, October 6th-9th, 2013*. Ed. by Michael F. Zaeh. Cham: Springer International Publishing, 2014, pp. 261–266. ISBN: 978-3-319-02054-9. DOI: 10.1007/978-3-319-02054-9_44.
- [50] Colin S. Harrison. „A review of automation in manufacturing illustrated by a case study on mixed-mode hot forging.” In: *Manufacturing Review* 1 (2014), p. 15. DOI: 10.1051/mfreview/2014012.
- [51] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren. „Reconfigurable manufacturing systems: Key to future manufacturing.” In: *Journal of Intelligent Manufacturing* 11.4 (2000), pp. 403–419. ISSN: 1572-8145. DOI: 10.1023/A:1008930403506.
- [52] Kevin MacG. Adams. *Non-functional requirements in systems analysis and design*. Springer International Publishing Switzerland, 2015.
- [53] N F O Evbuomwan, S Sivaloganathan, and A Jebb. „A Survey of Design Philosophies, Models, Methods and Systems.” In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 210.4 (1996), pp. 301–320. DOI: 10.1243/PIME\PROC\1996\210\123\02.
- [54] Tetsuo Tomiyama and Wouter Schotborgh. „Yet another model of design synthesis.” In: *Proceedings of the International Conference on Engineering Design ICED 2007*. 2007.
- [55] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote. *Engineering design, a systematic approach, 3rd ed.* Ed. by Ken Wallace and Lucienne Blessing. Springer Verlag London Limited, 2007.
- [56] INCOSE. *INCOSE systems engineering handbook*. Ed. by Cecilia Haskins. International Council on Systems Engineering/INCOSE, 2006.
- [57] Benjmanin S. Blanchard and Wolter Fabrycky. *Systems Engineering and Analysis, 5th ed.* Pearson Education Limited, 2014.
- [58] Mark W Maier and Eberhardt Rechtin. *The art of systems architecting, second ed.* CRC Press LLC, 2002.
- [59] Claus Ballegaard Nielsen, Peter Gorm Larsen, John Fitzgerald, Jim Woodcock, and Jan Peleska. „Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions.” In: *ACM Comput. Surv.* 48.2 (2015), 18:1–18:41. ISSN: 0360-0300. DOI: 10.1145/2794381.
- [60] G. Muller and J. H. Andersen. „Factory production line as SoS; a case study in airplane engine component manufacturing.” In: *2015 10th System of Systems Engineering Conference (SoSE)*. 2015, pp. 42–46. DOI: 10.1109/SYSOSE.2015.7151905.

- [61] Carin Rösiö. „Supporting the design of reconfigurable production systems.” PhD thesis. Mälardalen University, 2012.
- [62] Natalia Svensson Harari. „Design process of flexible assembly systems.” PhD thesis. Mälardalen University, 2015.
- [63] James Wallace Duda. „A decomposition-based approach to linking strategy, performance measurement, and manufacturing system design.” PhD thesis. Massachusetts Institute of Technology, 2000.
- [64] James S. Noble and J. M. A. Tanchoco. „Design justification of manufacturing systems—a review.” In: *International Journal of Flexible Manufacturing Systems* 5.1 (1993), pp. 5–25. ISSN: 1572-9370. DOI: 10.1007/BF01328737.
- [65] Ningxuan Kang, Cong Zhao, Jingshan Li, and John A. Horst. „A Hierarchical structure of key performance indicators for operation management and continuous improvement in production systems.” In: *International Journal of Production Research* 54.21 (2016), pp. 6333–6350.
- [66] Ram S. Sriram. „Accounting information system issues of FMS.” In: *Integrated Manufacturing Systems* 6.1 (1995), pp. 35–40. DOI: 10.1108/09576069510076153.
- [67] Krishan M. Gupta and A. Gunasekaran. „Costing in new enterprise environment: A challenge for managerial accounting researchers and practitioners.” In: *Managerial Auditing Journal* 20.4 (2005), pp. 337–353. DOI: 10.1108/02686900510592034.
- [68] Wulff Plinke, Mario Rese, and B. Peter Utzig. *Industrielle Kostenrechnung*. Springer-Verlag Berlin Heidelberg, 2015.
- [69] Bin Wu. „Manufacturing strategy analysis and system design-the complete cycle within a computer-aided design environment.” In: *IEEE Transactions on Robotics and Automation* 16.3 (2000), pp. 247–258. ISSN: 1042-296X. DOI: 10.1109/70.850643.
- [70] H.-P. Wiendahl, H.A. ElMaraghy, P. Nyhuis, M.F. Zäh, H.-H. Wiendahl, N. Duffie, and M. Brieke. „Changeable Manufacturing - Classification, Design and Operation.” In: *CIRP Annals* 56.2 (2007), pp. 783–809. ISSN: 0007-8506. DOI: <https://doi.org/10.1016/j.cirp.2007.10.003>.
- [71] J. S. Dahmann and K. J. Baldwin. „Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering.” In: *2008 2nd Annual IEEE Systems Conference*. 2008, pp. 1–7. DOI: 10.1109/SYSTEMS.2008.4518994.
- [72] G. Muller. „Are stakeholders in the constituent systems SoS aware? Reflecting on the current status in multiple domains.” In: *2016 11th System of Systems Engineering Conference (SoSE)*. 2016, pp. 1–5. DOI: 10.1109/SYSOSE.2016.7542899.
- [73] Attila Chikán. „Integration of production and logistics – in principle, in practice and in education.” In: *International Journal of Production Economics* 69.2 (2001). Strategic Planning for Production Systems, pp. 129–140. ISSN: 0925-5273. DOI: [https://doi.org/10.1016/S0925-5273\(99\)00102-4](https://doi.org/10.1016/S0925-5273(99)00102-4).
- [74] Ann-Louise Andersen, Thomas Ditlev Brunoe, Kjeld Nielsen, and Carin Rösiö. „Towards a generic design method for reconfigurable manufacturing systems: Analysis and synthesis of current design methods and evaluation of supportive tools.” In: *Journal of Manufacturing Systems* 42.Supplement C (2017), pp. 179–195. ISSN: 0278-6125. DOI: <https://doi.org/10.1016/j.jmsy.2016.11.006>.

- [75] Phillip Schrieverhoff. „Valuation of Adaptability in System Architecture.” PhD thesis. Technische Universität München, 2014.
- [76] Mark Sh. Levin. *Modular system design and evaluation*. Springer International Publishing Switzerland, 2015.
- [77] Bob Wielinga and Guus Schreiber. „Configuration-Design Problem Solving.” In: *IEEE Expert: Intelligent Systems and Their Applications* 12.2 (1997), pp. 49–56. ISSN: 0885-9000. DOI: 10.1109/64.585104.
- [78] U. Heisel and M. Meitzner. „Progress in Reconfigurable Manufacturing Systems.” In: *Reconfigurable Manufacturing Systems and Transformable Factories*. Ed. by Anatoli I. Dashchenko. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 47–62. ISBN: 978-3-540-29397-2. DOI: 10.1007/3-540-29397-3_4.
- [79] John S. Gero. „Design Prototypes: A Knowledge Representation Schema for Design.” In: *AI Magazine* 11.4 (1990), pp. 26–36. ISSN: 0738-4602.
- [80] Ann-Louise Andersen, Kjeld Nielsen, and Thomas Ditlev Brunoe. „Prerequisites and Barriers for the Development of Reconfigurable Manufacturing Systems for High Speed Ramp-up.” In: *Procedia CIRP* 51.Supplement C (2016). 3rd ICRM 2016 International Conference on Ramp-Up Management, pp. 7–12. ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2016.05.043>.
- [81] Vasdev Malhotra, Tilak Raj, and Ashok Arora. „Evaluation of Barriers Affecting Reconfigurable Manufacturing Systems with Graph Theory and Matrix Approach.” In: *Materials and Manufacturing Processes* 27.1 (2012), pp. 88–94. DOI: 10.1080/10426914.2011.551963.
- [82] Helena Forslund and Patrik Jonsson. „The impact of forecast information quality on supply chain performance.” In: *International Journal of Operations & Production Management* 27.1 (2007), pp. 90–107. DOI: 10.1108/01443570710714556.
- [83] Ian N. Durbach and Theodor J. Stewart. „Modeling uncertainty in multi-criteria decision analysis.” In: *European Journal of Operational Research* 223.1 (2012), pp. 1–14. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2012.04.038>.
- [84] NIST. „Combinatorial analysis - set partitions: stirling numbers.” In: *National Institute of Standards and Technology - Digital library of mathematical functions* (2017).
- [85] Y. Koren. „Reconfigurable Manufacturing Systems and Transformable Factories.” In: *Reconfigurable Manufacturing Systems and Transformable Factories*. Ed. by Anatoli I. Dashchenko. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. Chap. General RMS Characteristics. Comparison with Dedicated and Flexible Systems, pp. 27–45. ISBN: 978-3-540-29397-2. DOI: 10.1007/3-540-29397-3_3.
- [86] Sylvie Després and Camille Rosenthal-Sabroux. „Designing decision support systems and expert systems with a better end-use involvement: a promising approach.” In: *European Journal of Operational Research* 61.1 (1992). IFORS-SPC Conference on Decision Support Systems, pp. 145–153. ISSN: 0377-2217. DOI: [https://doi.org/10.1016/0377-2217\(92\)90276-F](https://doi.org/10.1016/0377-2217(92)90276-F).
- [87] Andrew Kusiak. *Computational Intelligence in Design and Manufacturing*. 1st. New York, NY, USA: John Wiley & Sons, Inc., 2000. ISBN: 0471348791.
- [88] K P Tripathi. „A Review on Knowledge-based Expert System: Concept and Architecture.” In: *International Journal of Computer Applications; Special issue on Artificial Intelligence Techniques - Novel Approaches & Practical Applications* (2011).

- [89] Alexandre Gachet. „A Framework for Developing Distributed Cooperative Decision Support Systems – Inception Phase.” In: *Proceedings of the 4th Informing Science Conference 2001*. 2001.
- [90] P. Hättenschwiler. *Neue Konzepte der Entscheidungsunterstützung*. Tech. rep. University of Fribourg, 1999.
- [91] Ching-Lai Hwang; Abu Syed Md. Masud. *Multiple objective decision making - methods and applications*. Springer-Verlag Berlin Heidelberg New York, 1979.
- [92] Adam Michael Ross. „Managing unarticulated value - changeability in multi-attribute tradespace exploration.” PhD thesis. Massachusetts Institute of technology, 2006.
- [93] D. Navinchandra. *Exploration and innovation in design - towards a computational model*. Springer-Verlag New York Inc., 1991.
- [94] F. Moya, J. M. Moya, and J. C. Lopez. „Evaluation of design space exploration strategies.” In: *Proceedings 25th EUROMICRO Conference. Informatics: Theory and Practice for the New Millennium*. Vol. 1. 1999, 472–476 vol.1. DOI: 10.1109/EURMIC.1999.794512.
- [95] Moritz Vaillant. „Design Space Exploration zur multikriteriellen Optimierung elektrischer Sportwagenantriebsstränge.” PhD thesis. Karlsruher Institut für Technologie, 2016.
- [96] Eunsuk Kang, Ethan Jackson, and Wolfram Schulte. „An Approach for Effective Design Space Exploration.” In: *Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems: 16th Monterey Workshop 2010, Redmond, WA, USA, March 31- April 2, 2010, Revised Selected Papers*. Ed. by Radu Calinescu and Ethan Jackson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 33–54. ISBN: 978-3-642-21292-5. DOI: 10.1007/978-3-642-21292-5_3.
- [97] Wouter O. Schotborgh. „Knowledge engineering for design Automation.” PhD thesis. University of Twente, 2009.
- [98] Ivan Mistrik; Rami Bahsoon; Rick Kazman; Yuanyuan Zhang, ed. *Economics-driven software architecture*. Elsevier Inc., 2014.
- [99] Steffen Nordahl Joergensen. „Developing modular manufacturing system architectures.” PhD thesis. Aalborg University, 2013.
- [100] Ann-Louise Andersen. „Development of changeable and reconfigurable manufacturing systems.” PhD thesis. Aalborg University, 2017.
- [101] Bin Wu. „An overview of the technical requirements for an integrated computer-aided manufacturing system design environment.” In: *International Journal of Manufacturing System Design* 2.1 (1995), pp. 61–72.
- [102] Felix T.S. Chan, Bing Jiang, and Nelson K.H. Tang. „The development of intelligent decision support tools to aid the design of flexible manufacturing systems.” In: *International Journal of Production Economics* 65.1 (2000), pp. 73 –84. ISSN: 0925-5273. DOI: [https://doi.org/10.1016/S0925-5273\(99\)00091-2](https://doi.org/10.1016/S0925-5273(99)00091-2).
- [103] Felix T.S. Chan and H.K. Chan. „Design of a PCB plant with expert system and simulation approach.” In: *Expert Systems with Applications* 28.3 (2005), pp. 409 –423. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2004.12.002>.
- [104] Hédi Chtourou, Wassim Masmoudi, and Aref Maalej. „An expert system for manufacturing systems machine selection.” In: *Expert Systems with Applications* 28.3 (2005), pp. 461 –467. ISSN: 0957-4174. DOI: <https://doi.org/10.1016/j.eswa.2004.12.007>.

- [105] M K Khan, I Hussain, and S Noor. „A Knowledge Based Methodology for Planning and Designing of a Flexible Manufacturing System (FMS).” In: *International Journal of advanced manufacturing systems* 13.1 (2011).
- [106] B. Malakooti and A. Kumar. „A knowledge-based system for solving multi-objective assembly line balancing problems.” In: *International Journal of Production Research* 34.9 (1996), pp. 2533–2552. DOI: 10.1080/00207549608905043.
- [107] Joseph M. Mellichamp, O-Joung Kwon, and Ahmed F. A. Wahab. „FMS Designer: An expert system for flexible manufacturing system design.” In: *International Journal of Production Research* 28.11 (1990), pp. 2013–2024. DOI: 10.1080/00207549008942849.
- [108] G. Michalos, A. Fysikopoulos, S. Makris, D. Mourtzis, and G. Chryssolouris. „Multi criteria assembly line design and configuration – An automotive case study.” In: *CIRP Journal of Manufacturing Science and Technology* 9.Supplement C (2015), pp. 69–87. ISSN: 1755-5817. DOI: <https://doi.org/10.1016/j.cirpj.2015.01.002>.
- [109] Harish A. Rao and P. Gu. „Expert self-organizing neural network for the design of cellular manufacturing systems.” In: *Journal of Manufacturing Systems* 13.5 (1994), pp. 346–358. ISSN: 0278-6125. DOI: [https://doi.org/10.1016/0278-6125\(94\)P2584-2](https://doi.org/10.1016/0278-6125(94)P2584-2).
- [110] Yue H. Yin, Andrew Y.C. Nee, S.K. Ong, Jian Y. Zhu, Pei H. Gu, and Lien J. Chen. „Automating design with intelligent human-machine integration.” In: *CIRP Annals* 64.2 (2015), pp. 655–677. ISSN: 0007-8506. DOI: <https://doi.org/10.1016/j.cirp.2015.05.008>.
- [111] Joseph M. Mellichamp and Ahmed F. A. Wahab. „An expert system for FMS design.” In: *Simulation* 48.5 (1987), pp. 201–208. DOI: 10.1177/003754978704800504.
- [112] Nancy Hyer and Urban Wemmerlöv. *Reorganizing the Factory*. Productivity Press, 2002.
- [113] X. Delorme, O. Batta'ia, and A. Dolgui. „Multi-objective Approaches for Design of Assembly Lines.” In: *Applications of Multi-Criteria and Game Theory Approaches: Manufacturing and Logistics*. Ed. by Lyes Benyoucef, Jean-Claude Hennet, and Manoj Kumar Tiwari. London: Springer London, 2014, pp. 31–56. ISBN: 978-1-4471-5295-8. DOI: 10.1007/978-1-4471-5295-8_2.
- [114] Alexandre Gachet and Pius Haettenschwiler. „Developing Intelligent Decision Support Systems: A Bipartite Approach.” In: *Knowledge-Based Intelligent Information and Engineering Systems: 7th International Conference, KES 2003, Oxford, UK, September 2003. Proceedings, Part II*. Ed. by Vasile Palade, Robert J. Howlett, and Lakhmi Jain. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 87–93. ISBN: 978-3-540-45226-3. DOI: 10.1007/978-3-540-45226-3_13.
- [115] Robbert-Jan Koebrugge. „Towards a smart synthesis tool for future assembly cell concepts.” MA thesis. University of Twente, 2015.
- [116] Martin Karelse. „Reconfigurability - a case study on designing modular assembly facilities for synthesis application.” MA thesis. University of Twente, 2015.
- [117] Martin Griffiths and István Mező. „A Generalization of Stirling Numbers of the Second Kind via a Special Multiset.” In: *Journal of Integer Sequences* 13 (2010).
- [118] Robert K Yin. *Case study research, fourth ed.* SAGE Publications, Inc., 2009.

- [119] Matthew B. Miles; A. Michael Huberman; Johnny Saldana. *Qualitative data analysis*. SAGE Publishing, Inc., 2014.
- [120] ATLAS.ti. „Atlas.ti qualitative data analysis; <http://atlasti.com/de/>.” In: (2017).
- [121] Amaresh Chakrabarti and Thomas P. Bligh. „An approach to functional synthesis of mechanical design Concepts: Theory, applications, and emerging research issues.” In: *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10.4 (1996), pp. 313–331. DOI: 10.1017/S0890060400001645.
- [122] Judea Pearl. *Heuristics - Intelligent search strategies for computer problem solving*. Addison-Wesley Publishing Company, Inc., 1984.
- [123] Marcello Colledani, Dávid Gyulai, László Monostori, Marcello Urgo, Johannes Unglert, and Fred Van Houten. „Design and management of reconfigurable assembly lines in the automotive industry.” In: *CIRP Annals* 65.1 (2016), pp. 441–446. ISSN: 0007-8506. DOI: <https://doi.org/10.1016/j.cirp.2016.04.123>.
- [124] Massimo Manzini, Johannes Unglert, Dávid Gyulai, Marcello Colledani, Juan Manuel Jauregui-Becker, László Monostori, and Marcello Urgo. „An integrated framework for design, management and operation of reconfigurable assembly systems.” In: *Omega* (2017). ISSN: 0305-0483. DOI: <https://doi.org/10.1016/j.omega.2017.08.008>.
- [125] Steven Haveman. „Communicating behavior of systems.” PhD thesis. University of Twente, 2015.
- [126] Abel van der Windt. „A Computational Design Synthesis Tool for Exploring Supply Chain Configurations.” MA thesis. University of Twente, 2017.
- [127] W. Ernst Eder. „Engineering design science and theory of technical systems: legacy of Vladimir Hubka.” In: *Journal of Engineering Design* 22.5 (2011), pp. 361–385. DOI: 10.1080/09544828.2010.522558.
- [128] James J Thomas and Kristin A Cook, eds. *Illuminating the path - the research and development agenda for visual analytics*. IEEE CS Press, 2005.
- [129] Robert Schmitt. *Cyber Physical Production Systems: Ohne Sensorik keine Industrie 4.0*. Tech. rep. RWTH Aachen/Fraunhofer IPT, 2014.
- [130] Paul Daugherty, Prith Banerjee, Walid Negm, and Allan E Alter. *Driving Unconventional Growth through the Industrial Internet of Things*. Tech. rep. Accenture, 2015.
- [131] Yang Lu. „Industry 4.0: A survey on technologies, applications and open research issues.” In: *Journal of Industrial Information Integration* 6.Supplement C (2017), pp. 1–10. ISSN: 2452-414X. DOI: <https://doi.org/10.1016/j.jii.2017.04.005>.
- [132] Davy Preuveneers and Elisabeth Ilie-Zudor. „The intelligent industry of the future: A survey on emerging trends, research challenges and opportunities in Industry 4.0.” In: *Journal of Ambient Intelligence and Smart Environments* 9.3 (2017), pp. 287–298. DOI: 10.3233/AIS-170432.
- [133] C. Scheuermann, S. Verclas, and B. Bruegge. „Agile Factory - An Example of an Industry 4.0 Manufacturing Process.” In: *2015 IEEE 3rd International Conference on Cyber-Physical Systems, Networks, and Applications*. 2015, pp. 43–47. DOI: 10.1109/CPSNA.2015.17.

- [134] V. Paelke. „Augmented reality in the smart factory: Supporting workers in an industry 4.0. environment.” In: *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*. 2014, pp. 1–4. DOI: 10.1109/ETFA.2014.7005252.
- [135] Hoda A. ElMaraghy and H.-P. Wiendahl. „Changeability – An Introduction.” In: *Changeable and Reconfigurable Manufacturing Systems*. Ed. by Hoda A. ElMaraghy. London: Springer London, 2009, pp. 3–24. ISBN: 978-1-84882-067-8. DOI: 10.1007/978-1-84882-067-8_1.
- [136] W. Terkaj, T. Tolio, and A. Valente. „Focused Flexibility in Production Systems.” In: *Changeable and Reconfigurable Manufacturing Systems*. Ed. by Hoda A. ElMaraghy. London: Springer London, 2009, pp. 47–66. ISBN: 978-1-84882-067-8. DOI: 10.1007/978-1-84882-067-8_3.
- [137] Yunus Kathawala. „Applications of Quantitative Techniques in Large and Small Organizations in the United States: An Empirical Analysis.” In: *Journal of the Operational Research Society* 39.11 (1988), pp. 981–989. ISSN: 1476-9360. DOI: 10.1057/jors.1988.168.
- [138] Scott M Shafer and Timothy L Smunt. „Empirical simulation studies in operations management: context, trends, and research opportunities.” In: *Journal of Operations Management* 22.4 (2004). *Simulation Studies in Operations Management*, pp. 345 – 354. ISSN: 0272-6963. DOI: <https://doi.org/10.1016/j.jom.2004.05.002>.
- [139] Pär Klingstam and Per Gullander. „Overview of simulation tools for computer-aided production engineering.” In: *Computers in Industry* 38.2 (1999), pp. 173 –186. ISSN: 0166-3615. DOI: [https://doi.org/10.1016/S0166-3615\(98\)00117-1](https://doi.org/10.1016/S0166-3615(98)00117-1).
- [140] L. M. S. Dias, A. A. C. Vieira, G. A. B. Pereira, and J. A. Oliveira. „Discrete simulation software ranking 2014; A top list of the worldwide most popular and used tools.” In: *2016 Winter Simulation Conference (WSC)*. 2016, pp. 1060–1071. DOI: 10.1109/WSC.2016.7822165.
- [141] A. Ingemansson; G.S. Bolmsjö; U. Harlin. „A Survey of the Use of the Discrete-Event Simulation in Manufacturing Industry.” In: *Proceedings of the 10th International Manufacturing Conference in China IMCC2002*. 2002.
- [142] Bo K. Wong, John K.S. Chong, and Jaesun Park. „Utilization and Benefits of Expert Systems in Manufacturing: A Study of Large American Industrial Corporations.” In: *International Journal of Operations & Production Management* 14.1 (1994), pp. 38–49. DOI: 10.1108/01443579410049298.
- [143] Öncü Hazir, Xavier Delorme, and Alexandre Dolgui. „A review of cost and profit oriented line design and balancing problems and solution approaches.” In: *Annual Reviews in Control* 40.Supplement C (2015), pp. 14 –24. ISSN: 1367-5788. DOI: <https://doi.org/10.1016/j.arcontrol.2015.09.001>.
- [144] Emanuel Falkenauer. „Line Balancing in the Real World.” In: *International Conference on Product Lifecycle Management*. 2005.
- [145] Denis Borenstein. „Intelligent decision support system for flexible manufacturing system design.” In: *Annals of Operations Research* 77.0 (1998), pp. 129–156. ISSN: 1572-9338. DOI: 10.1023/A:1018969126418.
- [146] Surya Raju Sagi and F. Frank Chen. „A framework for intelligent design of manufacturing cells.” In: *Journal of Intelligent Manufacturing* 6.3 (1995), pp. 175–190. ISSN: 1572-8145. DOI: 10.1007/BF00171446.

Part V

APPENDIX

A.1 SCENARIO AFTER REALIZING THE VISIONS OF INDUSTRY 4.0

The following scenario illustrates the potential benefits for companies, suppliers and customers that can be expected when combining various industry 4.0 technologies (based on [129, 130]).

Customers from all over the world use configuration tools available on the websites of the product manufacturers to customize the products to their specifications and place the orders online. In the manufacturer's plant, each customer order triggers the execution of its own chain of logistical and manufacturing processes: the products autonomously determine the needed manufacturing processes from raw materials to the final products, while the production systems autonomously configure themselves for the next product to be processed. Autonomous transport systems ensure that production can start with little delay, if not immediately, and help to keep waiting times for the customers to a minimum. While some processes are executed collaboratively by workers and robots, other processes are executed autonomously by the robots, supplying products cost-efficiently and at a consistently high quality. Employees are available to respond to the alerts that the systems automatically sends out when deviations from the normal operating mode are detected in the production system's live-monitored data. Based on these alerts, the workforce expertise is used to anticipate and respond to machine or system failures by performing corrective actions. The customers can track the progress of production in real-time from the order until the delivery to their location. Nevertheless, the products also automatically store this information, among other information such as the supplier and batch numbers of the raw materials, to provide full traceability in case problems show up later in the product's life.

A.2 EXCURSUS: INDUSTRY 4.0 STATE OF THE ART

Figure 68 shows the number of publications about the German initiative Industry 4.0 since its founding in 2011. Both the author's own research and the previous review by Lu [131], indicate the same trend: the quantity of papers dealing with Industry 4.0 is rising year by year, which suggests that the topic is still gaining popularity. While some authors identify challenges that research still has to overcome (e.g. [132]), others already present singular proof-of-concepts for cyber-physical production systems (e.g. [133]) or support technologies for designing CMS (e.g. [134]). On the one hand, these publications are a record of the progress and demonstrate that implementations are technically feasible. On the other hand, it remains unclear to which extent the industrial practice actually relies on such systems, since no empirical studies systematically investigated the application of CMS technologies in a broader range of companies and industries. While this can be an indication of the academic community lagging behind industrial developments, it can also indicate that the gap between vision and reality is substantial.

REVIEW METHODOLOGY Scopus-database search results with title containing the concepts "Industry 4.0" or "Industrie 4.0" considered from 2011 to 2016. Result was a total of 330 publications: 121 conference papers, 151 magazine articles, 51 journal publications and 6 unassigned publications that were excluded. For comparison, the results of Lu's review are presented [131].

Figure 68: Comparison of the number publications on Industrie 4.0/Industry 4.0

A.3 PRODUCTION SYSTEM CHANGEABILITY AND RELATED TERMINOLOGY

Researchers have pointed out various forms of flexibility with regard to the diverse system types (e.g. machines, transports systems, storage systems) and processes (production, transport, buffers) used in factories. Potential forms of flexibility are, for instance, machine flexibility (the ability to produce without requiring to change the tooling) and routing flexibility (the ability to choose different workstations for producing a product) [135]. To encapsulate the various forms of manufacturing flexibility, the overarching concept of changeability was introduced [70]. Figure 69 shows the distinct terms describing changeability in dependency of various levels at which production systems and products can be considered. On the lowest level, workstations can be changed over or different features of workpieces can be produced, which is referred to as changeover-ability. Extending this logic over the various levels leads to the highest form of flexibility, termed agility, where entire networks of factories or the product portfolio can be changed.

Figure 69: Changeability classification of production systems (based on [70])

Terkaj et al. approach changeability and the related terminology in the context of manufacturing systems from another perspective [136]. According to their definition, changeability can be expressed by structural, temporal and monetary characteristics of system changes and summarized in three main concepts: (1) flexibility is given if the system has a specific ability needed at a certain moment; (2) reconfigurability is present if the system is able to acquire the ability; (3) changeability allows to acquire the enabler for the system to become reconfigurable. These features can be illustrated on a concrete example: A system is flexible if it already has the production technology (e.g. spot welding) when a new product requires this type of process. In case the technology required for producing the new product is missing, but still can be integrated into the system, it can be considered reconfigurable. Lastly, if the system cannot integrate the required production technology immediately, but can be modified to make the integration possible, it is changeable. Hence, the concepts describe the readiness of the system and required efforts to react to changes: flexibility expresses immediate readiness and negligible effort; reconfigurability indicates that the provision of the ability can be slightly delayed and requires medium efforts; changeability can imply an even longer delay and high efforts; non-changeable systems cannot react at all and consequently make necessary to install additional systems in case the environment changes. The example relates the different terms – flexibility, reconfigurability and changeability –

to the specific system capability of executing a particular type of production process. Yet, the different concepts can be also applied to other properties of the production system, the so called flexibility dimensions [136]. Additional dimensions are the system's capacity, processes and production planning. Hence, the potential costs, duration and fashion of changing the production system represent three key variables that should be considered when designing flexible systems.

A.4 EXAMPLE OF PRODUCT PORTFOLIO AND SUITABLE SYSTEM TYPES

The table below illustrates a product portfolio, in which the production process requirements of the four products are partly similar (indicated steps i and ii; the Roman numerals indicate the number of the production step in the process sequence of the specific part), and partly heterogeneous. Hence, suitable production system designs could use the flow line principle for steps i (respective product parts 1,2,3) and ii (respective parts 1 and 2). The remaining processes could be either organized as a job-shop or as a cellular system. Yet, the context-specific details, such as processing durations of the products and their similarity, can promote applying the other principles.

Example of similar and dissimilar process sequences of four products

	<i>Machining processes of part Px</i>				<i>Assembly processes to component Cx and finished product FPx</i>			
	Milling	Drilling	Turning	Honing	Screwing	Welding	Gluing	Riveting
Prod. 1	P _{1,2,3} (i)	P _{1,2} (ii)	P ₃ (ii)		C ₁ :P ₁ +P ₂ (iii)	FP ₁ :C ₁ +P ₃ (iv)		
Prod. 2	P _{1,2,3} (i)	P _{1,2} (ii)		P ₃ (ii)	C ₁ :P ₁ +P ₂ (iii)		FP ₂ :C ₁ +P ₃ (iv)	
Prod. 3	P _{1,2,3} (i)	P _{1,2} (ii)	P ₃ (ii)	P ₃ (ii)	C ₁ :P ₁ +P ₂ (iii)			FP ₃ :C ₁ +P ₃ (iv)
Prod. 4	P _{1,2,3} (i)	P _{1,2} (ii)			C ₁ :P ₁ +P ₂ (iii)	FP ₄ :C ₁ +P ₃ (iv)		

A.5 EXCURSUS: INDUSTRIAL PRACTICE OF PRODUCTION SYSTEM DESIGN

Research focusing on the industrial practice of production system design is a valuable reference for identifying common approaches and the degree of application of computational support approaches.

APPLICATION OF METHODOLOGICAL PRODUCTION SYSTEM DESIGN Based on studies of the production system development process in ten manufacturing companies in Sweden, the researchers conclude that practice seems to be dominated by trial-and-error approaches with no systematic evaluations carried out [4]. Despite the disputability of these observations admitted by the authors, arising out of the applied, partly retrospective research approach, they stated that the results still clearly demonstrate that the way of designing is not in focus.

APPLICATION OF SIMULATION Considering the application of simulation in industrial practice, Chryssolouris found that system simulation was applied only in rare cases [23], however without providing empirical proof. Empirical studies found simulation applied in some production-related problems, such as design, planning and control and resource allocation [137, 138]. Yet, for interpreting the findings of the survey in the manufacturing companies described in [137], one should consider that it merely quantifies the availability of support tools in industry. Since the studies did not examine the actual use, the results might delude and imply a chance that the support tools are actually not used. An indication that simulation is an acknowledged support approach in industrial practice can be seen in the availability of commercial tools for performing simulations of production systems [139, 140]. However, also here the mere ability to purchase such tools does not allow to conclude that simulation is a standard approach in the design of production systems. A questionnaire-based survey of Swedish companies concluded that discrete event simulation was actively used in 12 out of the 80 surveyed companies [141], fostering the impression that models are applied, yet exceptionally. Since more recent literature reporting significant changes is difficult to find, the results of these surveys are assumed to be valid.

OPTIMIZATION APPROACHES Ambiguity is also present regarding the industrial application of optimization models in production system design. Empirical research conducted more than twenty years ago found that optimization approaches were partly applied in industry [142]. Nevertheless, also the results of this study might express the availability rather than the actual use of the techniques. A support of this argument can be seen by other researchers stating that optimization approaches are mainly results from academic research, yet often remain without practical application [23]. Supporting this view, a recently conducted review of optimization models for the design of production systems emphasized a lack of integration of the models into computational tools to enable industrial decision-makers to derive benefit from them [143]. Similarly, authors have pointed out difficulties practitioners face when intending to apply existing models, among which a lack of commercial applications, unsuitable assumptions of the models and a lack of efficiency [144–146].

After all, it is difficult to develop a clear impression of the general practice and use of support tools in industry out of two reasons. Firstly, the number of publications examining the production system design practice from an empirical point of view is low compared to the many prescriptive computational approaches that can be found in scholarly literature.

Secondly, the few studies found are difficult to compare as they range from detailed studies of the practice of individual companies to remote, questionnaire-based assessments of specific design process features in a large number of companies. While the representativeness of the results of the former study type might be difficult to interpret, the latter implies less controllable answers and consequently a potentially reduced validity. Based on the presented studies, the design of production systems in industrial practice appears to be little structured and only occasionally supported by computational approaches.

A.6 PRODUCTION SYSTEM MODEL DETAILS

The following shows the information describing each object class used in the model.

PRODUCT PROPERTIES

Product name	Product family	Transport cost rate	Average value	Production steps (<step 1>, <step 2>, ...)	Sequence constraints
Po1	1	9	250	<FAG1 - 15,0 - 2 >, <FAG2 - 6,0 - 1 >, <FAG3 - 8,0 - 2 >, <FAG1 - 5,0 - 3 >	<0 - 0 >
Po2	2	2	150	<FAG5 - 14,0 - 4 >, <FAG2 - 12,0 - 1 >, <FAG5 - 15,0 - 2 >, <FAG3 - 12,0 - 2 >	<0 - 0 >

- Product name
- Product family
- Transport cost rate: Cost of transports in the production system, e.g. from storage to a production cell, per product
- Average value: Arithmetic mean of raw material value and projected sales price.
- Sequence and specification of the production steps: Each step is defined by three elements < 1 - 2 - 3 > (1) the technology required (e.g. spot welding), (2) the number of processing units that the product requires (e.g. spot welds) and (3) the number of distinct tools required to execute the process (e.g. welding guns).
- Sequence constraints: Information specifying which production steps have to be performed in the same cell. <0 - 0> means that no constraints apply, whereas <1 - 2> indicates that the first and second step must be executed on the same cell.

DEMAND SCENARIOS

Scenario	Product	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇	Y ₈	Y ₉
Scen_0	Po1	1.200	960	768	615	492	394	316	253	203	163
Scen_0	Po2	1.200	960	768	615	492	394	316	253	203	163
Scen_1	Po1	1.080	864	692	554	444	356	285	228	183	147
Scen_1	Po2	1.080	864	692	554	444	356	285	228	183	147
Scen_2	Po1	972	778	623	499	400	320	256	205	164	132
Scen_2	Po2	972	778	623	499	400	320	256	205	164	132

- Scenario name
- Product name: Link to the technical specification of each product.

- Yearly demand volumes: Demand for each product throughout a number of years in the specified scenario.

PRODUCTION RESOURCES

Equipment name	Investment [€]	Length [mm]	Width [mm]	Time per unit [s]	Reconfiguration time [s]	Changeover time [s]
Turntable	60.000	3.000	3.000	1,0	3.000	300
Tool	3.000	1.500	1.500	0,0	1.000	330
ControlUnit	65.000	1.500	1.500	0,0	5.000	335
Robot	75.000	3.000	3.000	0,0	2.000	335
FAG1	80.000	2.500	2.720	1,0	40.000	240

- Equipment name
- Investment cost
- Length and Width: Physical dimensions required to calculate the required shop-floor space of the production machines and the system.
- Time per unit: Speed of the machine required to calculate the production process duration based on the number of processing units specified for each product.
- Reconfiguration time: Time needed to change the location of the production resource if the production system needs to be reconfigured.
- Changeover time: Time needed to change the tools of the production machine for producing a different product.

AS-IS PRODUCTION SYSTEM CONFIGURATION

Cell name	Equipment name	Allocation information
C01	FAG1	<P01 - 1,0 - 1 >, <P01 - 3,0 - 1 >
C01	FAG2	<P02 - 1,0 - 1 >

- Cell name
- Equipment name
- Allocation information: The production steps allocated to the equipment, specified by the product, the specific production step in the sequence of the product, and the number of resources used for completion the process.

PERFORMANCE ANALYSIS VARIABLES

Variable	Value
Yearly capacity	7.200.000 [s] (250 days with one 8-hour-shift)
System depreciation period length	5 [years]
Yearly depreciation interest rate	15%
Yearly interest rate on inventories	10%
EOQ ratio of setup to stock-holding cost	10

- Yearly capacity: Specification of the shift model for the system configuration.
- System depreciation period length: Time period in which the system equipment value is depreciated.
- Yearly depreciation interest rate: Interest rate for capital bound in equipment investments.
- Yearly interest rate on inventories: Interest rate for capital bound in stocks.
- EOQ ratio of setup to stock holding cost: The ratio of set-up to stockholding cost is a variable required to calculate the batch size of the products in the respective years based on the economic order quantity.

SYNTHESIZED PRODUCTION SYSTEM CONFIGURATIONS

⇒ See Section 4.2.4 and Appendix A.8

A.7 PARETO-EFFICIENCY FEATURE

The feature for identifying and visualizing Pareto-efficient solutions is part of the software's functionality for visualization and interaction and was designed by Hans Tragter. In the DSET, the Pareto-quality is as a normalized value that expresses the quality of a design in a multi-objective space by a single scalar value. A quality of 1,0 indicates that the solution is Pareto-efficient and values close to 1,0 are close to the Pareto-efficient points. Hence, the lower the value, the larger the distance from the Pareto-efficient designs. To calculate the Pareto-quality, an algorithm considers M solutions in the selected set. In the first step, each relevant performance P of the solutions is normalized, resulting in normalized scales for all relevant performances N . The Pareto-quality is then calculated for each solution i based on their averaged L1 distance – also known as the Manhattan distance (see magnified detail in Figure 70) – to the nearest Pareto-efficient point P_C , as shown in the following equation:

$$Q_i = 1 - \frac{\sum_{j=1}^N |P_{Cj} - P_{ij}|}{N} \quad (\text{A.1})$$

Figure 70 illustrates an example based on a screenshot from the DSET. Low values for two performances – investment and utilization – are selected for determining the Pareto quality. All Pareto efficient solutions are visualized as crosses instead of dots. The solution highlighted by the arrows is not fully efficient, however, is quite close to the closest Pareto-efficient solution (0,9393). Therefore the Pareto feature enables the users to detect all efficient solutions and quantify the distance of non-efficient solutions. In the illustrated example, solutions could be also visually evaluated in the 2D chart, yet, the Pareto-feature becomes important when assessing three or more solution properties simultaneously.

Figure 70: Example for Pareto quality feature

A.8 PROPERTIES AND ANALYSIS OF THE SYNTHESIZED SYSTEM CONFIGURATIONS

The following describes the system configuration's parameters and related analysis methods. For each product demand scenario, one set of the parameters presented below is stored.

SYNTHESIS-RELATED PROPERTIES

RUNID This textual description allows to identify system configurations created in the same synthesis run and therefore makes it possible to identify solutions created with the same set of requirements, e.g. on scaling factors.

PS NAME This parameter is a unique identifier for the generated production system in the format 'PS_number'. The systems are consecutively numbered based on the automatically executed synthesis attempts: if in five attempts to synthesize a solution, only number 1 and 4 result in a feasible system configuration, the names of the resulting system configurations will be "PS_1" and "PS_4".

DESIGN PROPERTIES

META An identifier for the design method applied to synthesize the configuration (e.g. 'ProductFamily').

SYSTEM HARDWARE DESIGN Textual description of the allocation of FAGs and other equipment components (described as CellBase) to the cells (e.g. Cell_1: FAG1(2x), FAG2(2x), FAG3(2x), FAG4(1x)/CellBase: 1 CU, 1 TT, 1 Rob, 11 Tools; Cell_2: FAG1(1x), FAG2(1x), FAG3(1x)/CellBase: 1 CU, 1 TT, 1 Rob, 6 Tools).

SYSTEM CONFIGURATION Textual description of the allocation of the products to the cells (e.g. Cell_1: P01 (Step 1), P02; Cell_2: P01 (Step 2 – 4), P03, P04).

ROUTING Textual description of the sequence of visited production cells for each product (e.g. P01: Cell 1, Cell 2; P02: Cell 1; P03: Cell 2; P04: Cell 2).

SCENARIO Identifier for the scenario to which the performances are related.

NUMBER OF CELLS Total number of production cells in the configuration, denoted by nC .

NUMBER OF FAGS Total number of FAGs in the configuration.

CELL SCALING FACTOR Value applied for the synthesis of the configuration.

FAG SCALING FACTOR Value applied for the synthesis of the configuration.

To facilitate interpreting the variables used in the following equations, the following table explains the meaning of the indices and parameter names.

Nomenclature of analysis parameters

Abbreviation	Description
$Index_C$	related to a cell; integrates all equipment in the cell
$Index_c$	related to changeover
$Index_d$	related to direct production
$Index_{Eq}$	related to an equipment component
$Index_{log}$	related to logistics
$Index_p$	related to a product
$Index_{rec}$	related to reconfiguration
$Index_{rep}$	related to reprogramming
$Index_S$	related to the system; integrates all cells in the system
$Index_{sto}$	related to storage
$Index_{tot}$	related to the total planning horizon
$Index_Y$	related to a year
A	Area
bth	Breadth
bs	Batch size
C	Cost
cr	Cost rate
d	Demand volume
EV	Equipment value
$ICTI$	Inter-cell transport indicator
I	Investment
L_{st}	Average stock-level
L_{tr}	Number of transports
n	number
nb	Number of batches
ncv	Number of cells visited
nps	Number of production steps
NUV	Not utilized value
r_{SSH}	Ratio of set-up and stockholding cost
T/t	Time
u	Utilization
W	Hourly wage of employee

AREA The only design property subject to analysis is the total shop floor area required by the system A_S , which is calculated based on the length and width required for placing the machine components in the system A_{Eq} :

$$A_{Eq} = lth \cdot bth \quad (A.2)$$

$$A_S = \sum_{eq=Eq_1}^{Eq_x} A_{eq} \quad (A.3)$$

ECONOMIC PROPERTIES

EQUIPMENT VALUE The equipment value is determined for each cell EV_C by adding up the investments of the equipment in the cell I_{eq} . Then the equipment value is determined for the entire system EV_S :

$$EV_C = \sum_{eq=Eq_1}^{Eq_x} I_{eq} \quad (A.4)$$

$$EV_S = \sum_{C=C_1}^{C_x} EV_C \quad (A.5)$$

INVESTMENT The investment I_S can be different from the equipment value, since equipment components may be already available in the As-Is configuration and therefore do not need to be purchased. Hence, the investment is determined by subtracting the equipment value of the reused components of the As-Is system configuration EV_{AIS} from the value of the synthesized configuration:

$$I_S = EV_S - EV_{AIS} \quad (A.6)$$

CELL COST RATE To account for the different sizes of the various production cells, the cost rate is determined for each cell to specify the value of a single unit of capacity during the depreciation period of the system of p_d years. Also, an interest rate i_d for the capital bound in the cell's equipment is considered. The available time for operating the cells per year T_y is used to determine the value per time unit cr_C . This cost rate is used to determine the reprogramming, changeover and direct production cost for each cell separately based on the time required by these activities. For each cell C :

$$cr_C = \frac{EV_C \cdot (1 + i_d)^{p_d}}{T_y \cdot p_d} \quad (A.7)$$

RECONFIGURATION COST To determine the reconfiguration cost C_{recS} , the difference between the synthesized system configuration and the As-Is configuration is analyzed. The time consumption for reconfiguring T_{recS} is estimated by considering all cells that need to be removed (duration t_{Cr} , number n_{Cr}) or added (t_{Ca} , n_{Ca}), as well as machines that are removed (t_{Mr} , n_{Mr}) from or added (t_{Ma} , n_{Ma}) to the production cells. Also, the machine spots that need to be recovered (t_{MSr} , n_{MSr}) and prepared (t_{MSp} , n_{MSp}) are taken into account. The resulting time consumption is translated into cost by considering the hourly wage of the employees (W).

$$T_{recS} = t_{Cr} \cdot n_{Cr} + t_{Ca} \cdot n_{Ca} + t_{Mr} \cdot n_{Mr} + t_{Ma} \cdot n_{Ma} + t_{MSr} \cdot n_{MSr} + t_{MSp} \cdot n_{MSp} \quad (A.8)$$

$$C_{recS} = T_{recS} \cdot (W + \sum_{c=C_1}^{C_x} cr_C) \quad (A.9)$$

REPROGRAMMING COST The reprogramming cost C_{rep} describes the cost caused by recalibrating the robots during the time T_{rep} after physically reconfiguring of the shop floor. This activity only needs to be performed once for each production step, hence the duration required per machine t_{rep} and the total number of production steps of all products $p \cdot nps$

is considered, as well as the number involved machines per step (nps_m ; normally one; maximum two). For each product P on the cell:

$$\text{if } \sum nps_m > \sum nps \rightarrow f_{PC} = \sum nps \quad (\text{A.10})$$

$$\text{else } \rightarrow f_{PC} = \sum nps_m \quad (\text{A.11})$$

For all products on the cell:

$$T_{repC} = t_{repC} \cdot \sum_{p=P_1}^{P_x} f_{PC} \quad (\text{A.12})$$

Resulting cost for each cell:

$$C_{repC} = T_{repC} \cdot cr_c \quad (\text{A.13})$$

Resulting cost in the system:

$$C_{repS} = \sum_{c=C_1}^{C_x} C_{repC} \quad (\text{A.14})$$

BATCH SIZES AND NUMBER OF BATCHES To be able to calculate the yearly capacity requirements of the various products, the cost for changeover needs to be considered, which is dependent on the number of batches produced of each product in each year. Therefore, the economic order quantity of each product bs_{P_y} is determined based on the demand in each year d_{P_y} to determine the number of batches nb_{P_y} for each year throughout the planning horizon of length y , considering the ratio of setup to stockholding cost r_{SSH} . For each product P (per cell c):

Batch-size of product P in year y:

$$bs_{P_y} = \sqrt{2 \cdot d_{P_y} \cdot r_{SSH}} \quad (\text{A.15})$$

Number of batches for product P in year y:

$$nb_{P_y} = \frac{d_{P_y}}{bs_{P_y}} \quad (\text{A.16})$$

DIRECT PRODUCTION AND CHANGEOVER COST An important assumption reflected in the calculations for determining the direct production time is that only one robot can be assigned to each cell, which completely executes the sequence of production steps for each product before continuing production of the next product of the batch. The durations of direct production T_{dC} and changeovers T_{cC} yield the direct production and changeover cost for each cell C_{dC} , C_{cC} and for the system C_d , C_c . Also the cycle times of the individual production steps t_{dPC_x} executed in the cell are taken into account, as well as the duration of changeovers t_{cC} and the number of resources to be changed over for the product nps_M .

please note: To improve the readability of the equations, the year index is not included in (Equation A.17) to (Equation A.26). Hence, these equations determine the resulting direct production and changeover cost in one year. The summarized production and changeover cost – considering all years in the planning horizon – is determined in (Equation A.27) and (Equation A.28).

Duration of production steps of product P:

$$t_{dPC} = \sum_{x=1}^{x=nps} t_{dPC_x} \quad (\text{A.17})$$

Duration of changeovers for product P:

$$t_{cPC} = np_{SM} \cdot t_{cC} \quad (A.18)$$

Production time required of product P:

$$T_{dPC} = \sum_{t=1}^y d_{p_t} \cdot t_{dPC} \quad (A.19)$$

Changeover time required of product P:

$$T_{cPC} = \sum_{t=1}^y nb_{p_t} \cdot t_{cPC} \quad (A.20)$$

For the cell c (considering all products): Production times required:

$$T_{dC} = \sum_{p=P_1}^{P_x} T_{dPC} \quad (A.21)$$

Changeover time required:

$$T_{cC} = \sum_{p=P_1}^{P_x} T_{cPC} \quad (A.22)$$

Direct production cost:

$$C_{dC} = T_{dC} \cdot cr_C \quad (A.23)$$

Changeover cost:

$$C_{cC} = T_{cC} \cdot cr_C \quad (A.24)$$

For the system (considering all cells) Direct production cost:

$$C_{dS} = \sum_{ci=C_1}^{C_x} C_{dC_{ci}} \quad (A.25)$$

Changeover cost:

$$C_{cS} = \sum_{ci=C_1}^{C_x} C_{cC_{ci}} \quad (A.26)$$

In the total planning horizon: Total direct production cost:

$$C_{dS_{tot}} = \sum_{t=1}^y C_{dS_t} \quad (A.27)$$

Total changeover cost:

$$C_{cS_{tot}} = \sum_{t=1}^y C_{cS_t} \quad (A.28)$$

NON-UTILIZED VALUE The non-utilized value of the system NUV_S represents the difference between the value of the equipment in the system and the cost caused by reconfiguration, reprogramming, direct production and changeovers during the depreciation period. This figure is supposed to enable an estimation of the provided excess capacity of the system and economically quantifies it.

$$NUV_S = \sum_{c=C_1}^{C_x} EV_{C_x} - \sum_{t=1}^{p_d} (C_{cS_t} + C_{dS_t}) - T_{recS} \cdot cr_C - C_{repS} \quad (A.29)$$

STORAGE COST The storage cost C_{sto} is determined by analyzing the number of cells visited by each product $ncvp$, the batch size in each year bs_{p_y} the average value of the product AV_p and the interest rate on capital bound in stocks i_{BC} . For each product:

$$C_{stoP} = \sum_{t=1}^y bs_{p_y} \cdot AV_p \cdot i_{BC} \cdot ncvp \quad (A.30)$$

For the system:

$$C_{sto} = \sum_{p=P_1}^{P_x} C_{stoP} \quad (A.31)$$

LOGISTICS COST The logistics cost C_{log} considers the transport cost rate per product cr_p . For each product:

$$C_{logP} = \sum_{t=1}^y 2 \cdot d_{p_y} \cdot cr_p \cdot ncvp \quad (A.32)$$

For the system:

$$C_{log} = \sum_{p=P_1}^{P_x} C_{logP} \quad (A.33)$$

TOTAL COST The total cost C_{total} is determined by adding up the cost for reconfiguration, reprogramming, changeovers, direct production, logistics and storage over the planning horizon.

$$C_{total} = C_{recS} + C_{repS} + C_{log} + C_{sto} + \sum_{t=1}^y (C_{dS_t} + C_{cS_t}) \quad (A.34)$$

TECHNICAL PROPERTIES

STOCK-LEVELS The average stock-levels L_{st} are calculated based on the average batch size of the product and express the average stock-levels throughout the planning period.

$$L_{st} = \frac{\sum_{p=P_1}^{P_x} \sum_{t=1}^y 0,5 \cdot bs_{p_y} \cdot ncvp}{y} \quad (A.35)$$

TRANSPORTS The number of transports L_{tr} is based on the number of batches that need to be transported to and from each visited cell for each product throughout the planning horizon. It quantifies the logistics effort caused by the specific allocation of products and resources.

$$L_{tr} = \sum_{p=P_1}^{P_x} \sum_{t=1}^y 2 \cdot nb_{p_y} \cdot ncvp \quad (A.36)$$

INTER-CELL TRANSPORT INDICATOR/ICTI The inter-cell transport indicator is another performance metric that expresses the logistics efforts by relating the number of batches for products that visit more than one cell nb_{IC} to the total number of batches of all products nb_{total} . The resulting ratio reflects the percentage of batches requiring multiple transports, indicating the complexity of material flows.

$$ICTI = \frac{nb_{IC}}{nb_{total}} \quad (A.37)$$

UTILIZATION / CELL-RELATED UTILIZATION / YEAR- AND CELL-RELATED UTILIZATION

The utilization rate expresses the ratio of the available time T_{total} to the time in which the cells are utilized due to reprogramming, production and changeovers. It can be expressed with reference to different time-frames and objects: for each cell and year (u_{C_y} ; Equation A.38), as arithmetic mean of each cell (u_C ; Equation A.39) or as arithmetic mean of all cells in the system (u_{total} ; Equation A.40). For each cell (year-related):

$$u_{C_y} = \frac{T_{\text{rep}} + T_d + T_c}{T_y} \quad (\text{A.38})$$

For each cell (average):

$$u_C = \frac{\sum_{t=1}^y u_{C_y}}{T_y \cdot y} \quad (\text{A.39})$$

For the system:

$$u_{\text{total}} = \frac{\sum_{c=C_1}^{C_x} u_C}{n_C} \quad (\text{A.40})$$

A.9 EXPERIMENTAL DATA

The data below exhibits the problem description used for experiment 1. The data for the experiments 2 to 12 is not presented, since the most relevant differences between the experiments are explained in the respective tables and text. To obtain the data for these experiments please refer directly to the author or the department.

PRODUCT DEFINITION

Product name	Product family	Transport cost rate	Average value	Production steps (<step 1>, <step 2>, ...)	Sequence constraints
Po1	1	9	250	<FAG1 - 9,0 - 1 >, <FAG2 - 15,0 - 1 >, <FAG1 - 5,0 - 1 >, <FAG4 - 9,0 - 3 >	<0 - 0 >
Po2	1	2	150	<FAG2 - 14,0 - 2 >, <FAG1 - 8,0 - 3 >, <FAG3 - 7,0 - 3 >, <FAG1 - 9,0 - 3 >	<0 - 0 >
Po3	2	9	120	<FAG3 - 10,0 - 3 >, <FAG1 - 9,0 - 1 >	<0 - 0 >
Po4	2	4	80	<FAG1 - 7,0 - 3 >, <FAG3 - 12,0 - 1 >, <FAG2 - 14,0 - 2 >	<0 - 0 >

SCENARIO DEFINITION

Scenario	Product	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9
Scen_0	Po1	1.200	960	768	615	492	394	316	253	203	163
	Po2	2.000	1.600	1.280	1.024	820	656	525	420	336	269
	Po3	3.800	3.040	2.432	1.946	1.557	1.246	997	798	639	512
	Po4	5.300	4.240	3.392	2.714	2.172	1.738	1.391	1.113	891	713
Scen_1	Po1	1.080	864	692	554	444	356	285	228	183	147
	Po2	1.800	1.440	1.152	922	738	591	473	379	304	244
	Po3	3.420	2.736	2.189	1.752	1.402	1.122	898	719	576	461
	Po4	4.770	3.816	3.053	2.443	1.955	1.564	1.252	1.002	802	642
Scen_2	Po1	972	778	623	499	400	320	256	205	164	132
	Po2	1.620	1.296	1.037	830	664	532	426	341	273	219
	Po3	3.078	2.463	1.971	1.577	1.262	1.010	808	647	518	415
	Po4	4.293	3.435	2.748	2.199	1.760	1.408	1.127	902	722	578

AS-IS SYSTEM CONFIGURATION DEFINITION

Cell name	Equipment name	Allocation information
C99*	FAG1	<P01 - 1,0 - 1 >, <P01 - 3,0 - 1 >
C99*	FAG2	<P02 - 1,0 - 1 >

* C99 means empty/not to be considered

EQUIPMENT DEFINITION

Equipment name	Investment [€]	Length [mm]	Width [mm]	Time per unit [s]	Reconfiguration time [s]	Changeover time [s]
Turntable	60.000	3.000	3.000	1,0	3.000	300
FAG1	15.000	2.000	2.000	1,0	18.000	1.800
FAG2	33.000	1.000	1.000	1,0	18.000	3.600
FAG3	250.000	10.000	1.000	1,0	18.000	4.500
FAG4	15.000	1.000	1.005	1,0	18.000	6.000
FAG5	30.000	1.750	1.755	1,0	7.000	325
Tool	3.000	1.500	1.500	0	1.000	330
ControlUnit	65.000	1.500	1.505	0	5.000	335
Robot	25.000	3.000	3.000	0	2.000	335

ANALYSIS VARIABLES

Variable	Value
Yearly capacity	7.200.000 [s]*
System depreciation period length	5 [years]
Yearly depreciation interest rate	15%
Yearly interest rate on inventories	10%
EOQ ratio of setup to stock-holding cost	10

* 250 days with one 8-hour-shift

A.10 EXAMPLE FOR CONFIGURATION OPPORTUNITIES

The first table presented below shows the possible FAG configurations of the two product families. Since $cSF=1$ only one cell per family is possible, however, since $fSF=2$, the cells of product family 1 can be configured in 16 different ways, and the cell of product family 2 in 8 ways.

Possible cell configurations of the two product families												
	Product family 1						Product family 2					
	Cell	Cell	Cell	...	Cell	Cell	Cell	Cell	Cell	Cell	Cell	Cell
	conf.	conf.	conf.		conf.	conf.	conf.	conf.	conf.	conf.	conf.	conf.
	ID ₁	ID ₂	ID ₃		ID ₁₅	ID ₁₆	ID ₁	ID ₂	ID ₃	ID ₇	ID ₈	
FAG ₁	1X	2X	1X	...	2X	2X	1X	2X	1X	...	2X	2X
FAG ₂	1X	1X	2X	...	2X	2X	1X	1X	2X	...	2X	2X
FAG ₃	1X	1X	1X	...	2X	2X	1X	1X	1X	...	1X	2X
FAG ₄	1X	1X	1X	...	1X	2X						

The following table illustrates the explosive growth of the combinatorial opportunities when changing the cell scaling factor to $cSF=2$. The total number of alternatives increases from 128 distinct system configurations possible with two cells to 60.544 configurations with two to four cells.

Possible system configurations of the two product families					
System configuration	Cell ₁	Cell ₂	Cell ₃	Cell ₄	Product of row
Cell ₁ $\Rightarrow$ Prod. fam. 1; Cell ₂ $\Rightarrow$ Prod. fam. 2	16	8			128
C ₁ $\Rightarrow$ PF 1; C ₂ $\Rightarrow$ PF 2; C ₃ $\Rightarrow$ PF 2	16	8	8		1.024
C ₁ $\Rightarrow$ PF 1; C ₂ $\Rightarrow$ PF 1; C ₃ $\Rightarrow$ PF 2	16	16	8		2.048
C ₁ $\Rightarrow$ PF 1; C ₂ $\Rightarrow$ PF 2; C ₃ $\Rightarrow$ PF 2; C ₄ $\Rightarrow$ PF 2	16	8	8	8	8.192
C ₁ $\Rightarrow$ PF 1; C ₂ $\Rightarrow$ PF 1; C ₃ $\Rightarrow$ PF 2; C ₄ $\Rightarrow$ PF 2	16	16	8	8	16.384
C ₁ $\Rightarrow$ PF 1; C ₂ $\Rightarrow$ PF 1; C ₃ $\Rightarrow$ PF 1; C ₄ $\Rightarrow$ PF 2	16	16	16	8	32.768
Total alternatives ($cSF = 2$; $fSF = 2$):					60.544

A.11 EVALUATION INTRODUCTION

The experts evaluating the DSET read the following introduction that was used to stage the experiment.

SITUATIONAL CONTEXT

Please read the short text below before using the tool.

COMPANY AND PRODUCTS You are an employee of MetalForm Incorporated (MFI), a supplier in the automotive industry. MFI produces assemblies from metal and aluminum and has a good reputation for delivering consistent quality of body parts (example shown in Figure 71, left), even though only small volumes per design are produced. MFI produces various product types, for instance trunks, hoods, fenders, doors and side panels for various models of different customers, which results in a high-mix, low-volume structure of the demand. The most commonly applied production technologies are visualized in Figure 71 on the right.

Figure 71: Example for products (left) and production processes (right)

THE PROBLEM In the last years the automotive market has extremely diversified the model range. This makes it more and more difficult to make profit, as the lifecycles of most models are getting shorter, the competition is tough and production volumes per model are not as high as in the past. Your department has developed a reconfigurable manufacturing system (see Figure 72). In this concept, the production technologies/machines are grouped together (picture on the right). The cells consist of modular hardware components that are grouped around a central track, on which a robot moves. The robot transports the parts inside the cell, whereas forklift transport the parts between the cells and the storage area. If the utilization of the cell is getting too low due to changing demand, the cells can be reconfigured: the hardware modules can be regrouped to yield new cell designs, which gives opportunities for improving the system’s performance. However, you as the system designers have to consider that allocating each product and production step will affect all three types of subsystems and their performances such as cell utilization, logistics movements and stock levels in storage.

DESIGN PROCESS CONTEXT

The design process you are responsible for is first step of a series of processes that should eventually yield the configuration of the production system to be implemented. Hence, you are supposed to perform the first design step and generate a concept. While the nature of the concepts in scope is already known and defined by the reconfigurable system

Figure 72: Reconfigurable production system concept

architecture, the exact grouping of the processing resources and production steps yet has to be decided. The input for the process is data describing the products to be produced and their expected demand volumes. Furthermore, the modular machines for the future system are listed in a table, as well as the description of the current system configuration. The system and analysis constants are listed in an additional table. The result of the design process should give an idea on how many cells you should produce and how to allocate the production resources and products.

POSSIBLE SYSTEM TYPES

Figure 73 illustrates the three possible configuration types which are created by the design-rule sets: allocations yielding cells based on product families, dedicated cells and universal cells.

TOOL USAGE

The tool visualizes the solutions that were created using the algorithmic design procedures from the knowledgebase. It should be noted that semi-random algorithms are used to vary parameter values between the applicable boundaries for these parameters. For instance, the algorithm determines the boundaries for the parameter “number of cells” based on the number of products. A typical result of this analysis would be a minimum of 1 and a maximum of 5 for the number of cells. Then, the algorithm chooses one random value from this range to create an instance of the design (e.g. 4). This results in a 1 in 5 chance to obtain a specific parameter value for a single design created. In case 100 solutions are created theoretically 20% of these solutions will have that characteristic, but not always. The random influence and resulting probabilities for obtaining a specific characteristic imply that it cannot be certainly stated if the generated solutions really represent all opportunities. If the number of designs with intended characteristics appears sparse, different strategies can be tested to increase the number, and eventually also quality of solutions. One possible strategy is described below, nevertheless, users may develop their own strategies for the considered problems.

Figure 73: Three possible configuration types

Potential strategies for using the tool

- Start without any constraints, assess the solution space and check if a new run is required. In case solutions with non-relevant property values are found, add a requirement that excludes these values.
- If no solutions can be found with the current set of requirements, try again.
- If repeatedly no solutions can be found, check if requirements can be relaxed.
- If the number of solutions within requirements is very high, use the pareto-shapes to visualize and identify the solutions with the intended properties.
- If you think you have identified the most favorable solutions, try to identify what they have in common and make another attempt to create solutions with requirements reflecting these commonalities.

A.12 SCENARIOS

The board of the company has developed a long-term vision that aims at becoming the supplier with the most competitive prices. Nevertheless, they also want to increase the profitability of the company and are willing to invest into your department if you present acceptable plans for the assembly department. They ask you to come up with at least three alternative opportunities for configuring the production system, including an evaluation of the effects for logistics and storage. Furthermore, the board wants to know for each of your proposals how much effort it will take to implement the specific configuration taking into account today's configuration (re-configuration). Last but not least, each solution has to fulfill a set of requirements.

SCENARIO A

The following specifications characterize the first scenario.

- 5 products
- 3 demand scenarios
- 5 FAG types, 4 types of support equipment (robot, control unit, tool, turntable)
- Solution requirements: one cell per product family; investment < 1.000.000; capacity reserve > 25%;

SCENARIO A — ADDITION I

After discussing with the management, you could negotiate a partial softening of the requirements. You are allowed to also consider other types of configurations (thus not limited to product families anymore) if they provide the 25% capacity reserve and you manage to keep the investment under 800.000€.

SCENARIO A — ADDITION II

Your sales representative tells you that the demand expectation for products of customer A has risen drastically (+30%), while customer B had a terrible launch of the product (-25%). Do you see any opportunity to consider this changed situation?

SCENARIO B

The following specifications characterize the second scenario.

- 20 products
- 3 demand scenarios
- 5 FAG types, 4 types of support equipment (robot, control unit, tool, turntable)
- Solution requirements: less than 10 cells; area < 300 m²; investment < 1.000.000; direct production cost < 2.500.000; capacity utilization as high as possible;

A.13 QUESTIONS ASKED DURING THE EVALUATIONS

The following questions were asked after the two rounds of experiments. Questions Q9 to Q19 were asked *mutatis mutandis*.

Questions evaluating the presence of the factors, answered after scenario A and additions. The customizability of the DSET to different problem formulations was not covered by questions, since it was implicitly demonstrated by the application to the two cases.

Q1a: Does the tool propose different configurations with regard to hardware arranging the hardware?

Q1b: Does the tool propose different configurations with regard to product allocation?

Q2a: Does the tool allow you to consider the requirements that were described for the solutions?

Q2b: Does this make the solutions found more feasible with regard to implementation?

Q3: Does the tool allow you to evaluate the performance sensitivity of configurations?

Q4: Does the tool allow you to compare solutions to each other with regard to various properties?

Q5: Does the tool support you in identifying promising solutions?

Q6: Does the automated proposal of configurations increase the number of solutions you evaluate?

Q7: Does the automated proposal in conjunction with the automated performance analysis have an effect on the speed of solution evaluation?

Q8: Do you consider the way of providing or changing problem information as efficient?

Questions for individually evaluating the factors' relation to the effectiveness (Q8 to Q13) and the efficiency (Q14-Q19) of the design process, answered after scenario B.

Q9: Does the automated proposal of configurations increase the number of solutions that can be evaluated?

Q10: Does the requirement-based filtering of solutions enable you to identify a higher number of solutions that are feasible with regard to implementation?

Q11: Do the features to compare the solutions enable you to identify better performing solutions?

Q12: Does the scenario-related performance of the solutions enable you to identify better performing solutions?

Q13: Does the ability to identify better performing solutions enable you to select the most suitable solutions?

Q14: Does the functionality of the DSET represent an effective approach to consider a high number of feasible target configurations?

Q15: Does the automated synthesis, analysis and requirement-based filtering of solutions

allow to efficiently identify feasible solutions?

Q16: Do the features to compare solutions enable to efficiently identify suitable solutions?

Q17: Does the scenario-related performance of the solutions enable you to efficiently assess their sensitivity?

Q18: Do the tool features for visualizing and recognizing configurations that perform better than others enable you to perform an efficient assessment of the design solutions?

Q19: Does the functionality of the DSET allow for an efficient approach to consider a high number of feasible target configurations, including the time required for preparing, generating, evaluating and selecting the configurations?

Q20: Does the functionality of the DSET enable you to consider the actual options for system configuration more exhaustively?

Questions for evaluating the course of the session and closing it:

Q21: What did you think of the session?

Q22: Did you find it useful? Why or why not?

Q23: Did you enjoy it? Why or why not?

Q24: What do you think of the duration of the sessions?

Q25: Would you want to do such a session more often? Why or why not?

Q26: If so, at which kind of moments, and at which intervals?

Q27: Do you have any suggestions for improvement?

Q28: Do you think the tool is clear and understandable? Why or why not?

Q29: What do you think of the mechanism to get to interesting solutions?

Q30: Are relevant aspects lacking that you would need to make a decision?

Q31: Are any criteria redundant or unnecessary? If so, which?

Q32: During the sessions, multiple solutions have emerged as the most suitable. Would you yourself have chosen for the same solution? Why or why not?

Q33: Do you feel that you now better understand how to reach a design that meets internal and external requirements? How come?

Q34: Which of the current stakeholders (internal and external) in the design process would need to be involved into the usage of the tool?

Q35: Is this approach to decision making suitable to your current processes?

Q36: Do you have any other remarks?

Questions for evaluating the session realism:

Q37: Is the information used in scenarios available at this stage in the process? (*asked respectively for input data, requirements and objectives; author's note*)

Q38: How realistic are the scenarios with respect to the input data, requirements and objectives?

A.14 EVALUATION CASE SOLUTIONS

The following two tables show the solutions of the scenarios A and A – addition I that were determined by the experts with the support of the DSET.

please note: Despite the thorough preparation of the experiments, the DSET prototype showed technical difficulties in connection with the dataset suggested for 'Scenario A'. Since the search for the error and its resolution appeared potentially time-intensive and would have caused a delay in the evaluation schedule, the dataset for scenario B was used to generate solutions. The different data basis might have had impact on the generated solutions and their properties, and hence, also the presented solutions that were selected by the experts. As this change mostly affected the various solution property values and did not affect the intended and observed usage of the DSET, the implications for the validity of the findings – the usefulness and applicability of the tool – can be considered negligible.

Solutions of Scenario A

	Solution 1	Solution 2	Solution 3
Solution ID	PS_5664	PS_8867	PS_9197
Configuration type	Product-family	Product-family	Product-family
Area	148	151	128
Equipment value	1.185.000	1.372.000	1.052.000
Investment	186.000	373.000	53.000
Total cost	14.107.890	14.497.652	14.406.174
Non-utilized value	603.308	649.161	648.346
Utilization	67,00%	61,20%	61,20%

Solutions of Scenario A — addition I

	Solution 1	Solution 2	Solution 3
Solution ID	PS_23540	PS_24071	PS_25112
Configuration type	Multi-cell	Multi-cell	Product-family
Area	215	232	155
Equipment value	1.128.000	1.145.000	1.387.000
Investment	129.000	146.000	388.000
Total cost	20.874.644	21.450.853	14.323.635
Non-utilized value	495.366	576.373	636.135
Utilization	48,60%	45,90%	64,60%