

# Modellbasierte Planung und Steuerung unter Unsicherheit

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur  
der technischen Wissenschaften**

im Rahmen des Studiums

**Wirtschaftsinformatik**

eingereicht von

**Daniel Fellner**  
Matrikelnummer 0307116

an der  
Fakultät für Informatik der Technischen Universität Wien

Betreuung:  
Betreuer: Univ.Prof. Mag. Dr. Walter Schwaiger

Wien, 18.10.2010

\_\_\_\_\_  
(Unterschrift Verfasser)

\_\_\_\_\_  
(Unterschrift Betreuer)

# I. Eidesstattliche Erklärung

Familienname: Daniel

Vorname: Fellner

Geburtsdatum: 15. April. 1984

Adresse: Kirchbichlstraße 3/16  
9400 Wolfsberg  
Österreich

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, am 20.10.2010

.....

## **II. Gender Statement**

Aus Gründen der Lesbarkeit und um die Regeln der deutschen Grammatik und Sprache nicht permanent verletzen zu müssen, gibt es in diesem Dokument keine Unterscheidung zwischen Studentinnen und Studenten oder andere ähnliche Unterscheidungen.

Wortendungen wie Student/in, Studenten und –innen, Student(en)/innen und dergleichen werden aus oben genannten Gründen in dieser Arbeit keine Verwendung finden. Grammatikalisch korrekte Unterscheidungen im Geschlecht kommen natürlich zur Anwendung.

Grundsätzlich sind die Termini geschlechtsneutral gedacht und so verwendet, wie sie die deutsche Sprache vorsieht und schon immer vorsah.

### **III. Danksagung**

Ich möchte allen Leuten, die mich bei meinem Studium unterstützt haben, danken. Allen voran meinen Eltern, die mir Ausbildung und Studium erst ermöglicht haben und mich auch in vielen anderen Belangen immer eine moralische Stütze waren und mir mit ihrem Rat und Tat zur Seite standen.

Ebenso danke ich den vielen Freunden und Kollegen, die einen Teil meines Weges mitgegangen sind, mit denen ich mich gemeinsam auf Prüfungen vorbereitet habe und auch viel Spaß hatte. Mein spezieller Dank gilt dabei folgenden an dieser Stelle namentlich erwähnten Freunden und Kollegen: Aylin, Christian, Babsi, Marcel, Michael, Markus, Steffi und Ziad. Noch viele andere wichtige Leute haben mich in meinem Studium begleitet, die ich um Verzeihung bitte, wenn ich sie nicht namentlich erwähne. Ich danke euch für die gute Zusammenarbeit, die gegenseitige Unterstützung und Freundschaft.

Dem Lehrkörper der TU Wien danke ich dafür, dass sie mein Studium interessant gestaltet haben und in einigen Vorlesungen den Stoff hervorragend und spannend vorgetragen haben. Einige von Ihnen haben mein Verständnis für die fächerübergreifenden Zusammenhänge in der Wirtschaftsinformatik geschärft oder erst mein Interesse dafür geweckt. In manchen Vorlesungen war die Begeisterung des Vortragenden für ihr Fach richtig spürbar und vor allem ansteckend, was eine ungemeine Erleichterung für die Vorbereitung auf Prüfungen war, da die Motivation für das Fach Eigeninteresse anstatt einer positive Note war.

Abschließend möchte ich mich besonders bei Univ.Prof. Mag. Dr. Walter Schwaiger bedanken, der mir die Möglichkeit bot, ein sehr interessantes Thema zu bearbeiten. Ich bedanke mich für die zahlreichen Orientierungshilfen in einem sehr komplexen Themengebiet und die vielen Diskussionen, auch wenn manche davon etwas hitziger von Statten gingen weil bei komplexen Themen nicht selten Missverständnisse entstanden. Besonders lobend zu erwähnen ist das Engagement, das er für meine Betreuung aufgebracht hat. Wann immer ich einen Wink in die richtige Richtung gebraucht habe, fand ich schnell ein offenes Ohr für meine Probleme. Besonders dankend sei auch Christian erwähnt, der meine Wissenslücken bei der Programmierung mit sehr viel Geduld und Erfahrung durch die passende Herangehensweise an diverse Probleme gefüllt hat.

## **IV. Präambel (Deutsch)**

Planung und Steuerung sind Teil jener Kernkompetenzen, ohne welche sich heutzutage kein Unternehmen ab einer bestimmten Größe mehr auf eine betriebswirtschaftlich sinnvolle Art und Weise führen lässt.

Ein Vehikel zur Kontrolle eines Unternehmens sind ERP-Systeme, welche die Ressourcen und Vorgänge innerhalb eines Unternehmens systematisch erfassen. In diesem Forschungsgebiet wurde die Idee für diese Arbeit geboren. Dabei wird nicht auf der Basis altbewährter und etablierter Konzepte, welche in den heute verbreiteten ERP-Systemen Anwendung finden, gearbeitet. Am Institut für Managementwissenschaften der TU Wien entsteht unter der Federführung von Univ.-Prof. Mag. Dr. Walter Schwaiger ein ERP-System namens ERP-Control, welches neue Ansätze verfolgt und somit Alternativen zu den Paradigmen vergangener Tage implementiert und erforscht.

Ein ERP-System beinhaltet auch Planungsfunktionalitäten. Mit Hilfe von Plänen versuchen Menschen sich für eine ungewisse Zukunft zu wappnen und sie so gleichsam vorweg zu nehmen. Das kann natürlich nur in einem bestimmten Rahmen passieren, da die Zukunft ja noch nicht fest steht und mit entsprechender zeitlicher Entfernung zur Gegenwart immer schwerer vorherzusagen ist. Der Umgang mit dieser Unsicherheit ist ein Themengebiet dieser Arbeit.

Das Mittel zur Realisierung dieser Funktionalität ist die stochastisch optimale Produktionsplanung und –steuerung (SOPPS). Diese soll über mehrere Perioden hinweg eine möglichst optimale Ausbringungsmenge für die Produktion ermitteln. Pläne bilden jedoch im Zeitverlauf die künftige Gegenwart immer ungenauer ab. Hier wird mit Steuermechanismen versucht, den Plan anzupassen und im Laufe mehrerer Zeitperioden als gutes Modell für die Realität zu kalibrieren. Ansätze dafür kommen aus der Kybernetik, der Wissenschaft, die sich mit Steuerungs- und Regelungsprozessen befasst. Aus dem kybernetischen Steuerungsprinzip wird in dieser Arbeit eine Methodik erarbeitet, welche auf Basis von Erwartungswerten die Produktionsmengen intertemporal optimiert steuert. Dabei wird die stochastische Kontrolltheorie angewandt, um den der Zukunft immanenten Zufall den Schrecken zu nehmen und in gewissen Schranken berechenbar zu machen.

## **V. Abstract (English)**

Nowadays Planning and Control are part of those core competencies, without a company of a certain size cannot be led in a economical and reasonable way.

Tools to lead a company are ERP-systems, which are collecting data about resources and processes of a company. In this field of research the idea for this thesis has become alive. At the Institute for Management Sciences at Vienna University of Technology a team under leadership of Univ.-Prof. Mag. Dr. Walter Schwaiger develops an ERP-system called ERP-Control. Instead of using established concepts implemented in popular ERP-systems ERP-Control realizes new concepts and gets rid of old paradigms.

An ERP-system also contains planning functionality. Due to plans people try to handle an uncertain future. This only can happen in a certain scale because we cannot know the future exactly and with an increasing time distance to present the uncertainty of the prediction is also increasing. Managing this uncertainty is one of the main topics of this thesis.

One way to realize planning functionality is stochastic optimal production planning and control. Across multiple periods it calculates the optimal output amount for production. Keep in mind that plans are only models of reality and are getting more and more inexact over time. To calibrate a plan over time control mechanisms are getting used to generate a good model for describing reality. This approach comes out of cybernetics, a science which is concerned in control topics. In this thesis a method based on principles of cybernetics gets developed, which is using expected values for an intertemporal control of production output. Using stochastic control theory as an additional approach tries to make uncertainties and coincidences computable in certain limits.

## VI. Inhaltsverzeichnis

I. Eidesstattliche Erklärung .....	i
II. Gender Statement .....	ii
III. Danksagung .....	iii
IV. Präambel (Deutsch).....	iv
V. Abstract (English) .....	v
VI. Inhaltsverzeichnis .....	vi
<b>1. Einführung .....</b>	<b>1</b>
1.1. Problemstellung .....	1
1.2. Ziel der Arbeit.....	2
1.3. Struktur der Arbeit .....	4
1.4. Evolution von ERP .....	5
<b>2. Überblick über ERP-Control .....</b>	<b>10</b>
2.1. Architektur eines ERP-Systems .....	10
2.2. ERP-Control .....	12
2.3. Abgrenzung zu anderen MIS-Systemen .....	14
2.4. Metamodell von ERP-Control.....	16
2.4.1. REA-Ontologie .....	16
2.4.2. Kybernetische Betrachtung eines Systems.....	18
2.4.3. Relevante Normen .....	22
2.4.3.1. ECSI-Standard (DIN EN 62264) .....	23
2.4.3.2. AEO-Standard (ISO/IEC 15944-4).....	29
2.4.3.3. COSO II .....	31
2.4.4. Relevante Konzepte.....	35
2.4.4.1. Strategische Planung .....	35
2.4.4.2. Balanced Score Card .....	36
2.4.4.3. Hierarchiemodell nach Mesarovič.....	37
2.5. Mathematische Grundlagen .....	39
2.5.1. Umsatzergebnisfunktion .....	39
2.5.2. Stochastische Prozesse und Filtrierte Wahrscheinlichkeitsräume .....	41
2.5.3. Rekursiver Lösungsansatz für das SOPPS .....	46

<b>3. Implementierung der SOPPS</b> .....	<b>50</b>
3.1. Konzeptionelle Überlegungen .....	50
3.2. Verwendete Technologien.....	55
3.2.1. Enterprise Java Beans.....	55
3.2.2. Hibernate .....	57
3.2.3. JavaServer Faces .....	58
3.2.4. JBoss Seam.....	60
3.3. Prototyp der SOPPS .....	61
3.3.1. Datenstrukturen .....	62
3.3.1.1. ProductionPlanModelEntity-Klasse.....	62
3.3.1.2. Node-Klasse .....	64
3.3.2. Auszüge der Geschäftslogik .....	65
3.3.2.1. Generierung der Baumstruktur .....	65
3.3.2.2. Kalibrierung des auf die Baumstruktur aufgesetzten stochastischen Prozess .....	67
3.3.3. Prozessorientierte Benutzeroberfläche .....	74
3.3.3.1. Generierung und Modifikation des Produktionsplans .....	75
3.3.3.2. Genehmigung des Plans .....	82
3.3.3.3. Simulation und Durchführung des Plans.....	84
<b>4. Resümee</b> .....	<b>87</b>
4.1. Analyse des Lösungsansatzes.....	87
4.1.1. Leistungsumfang.....	87
4.1.2. Vor- und Nachteile der Lösung .....	90
4.1.3. Offene Fragen und ungelöste Probleme .....	92
4.2. Zusammenfassung.....	93
4.3. Blick in die Kristallkugel.....	94
4.4. Persönliches Resümee .....	95
A. Abkürzungsverzeichnis .....	a
B. Notationsverzeichnis.....	b
C. Abbildungsverzeichnis .....	c
D. Tabellenverzeichnis .....	d
E. Formelverzeichnis.....	d
F. Codeverzeichnis .....	d
G. Literaturverzeichnis.....	e



# 1. Einführung

## 1.1. Problemstellung

Nur wenige Unternehmen finden sich heutzutage in der glücklichen Lage wieder ein Monopol am jeweiligen Marktsektor auszufüllen. Die meisten Unternehmen sehen sich mit einer starken Konkurrenz konfrontiert und die Kunden reagieren natürlich dementsprechend. Sie wählen aus dem für sie am besten geeigneten Produkt und die Konsequenz daraus sind zahlreiche Unsicherheiten für die einzelnen Unternehmen. Es wird immer schwieriger Nachfragemengen abzuschätzen, wodurch die Produktionsplanung mitunter auf einem unsicheren Fundament steht. Daraus ergeben sich diverse Schwierigkeiten und Probleme, die es zu überwinden gilt.

Kommt es zu einer Überproduktion und ein Unternehmen produziert mehr als es absetzen kann, fallen Nebenkosten z.B. in der Form von Lagerkosten an. Anstatt dass das Produkt Geld einbringt, kostet es stattdessen dem Unternehmen etwas. Das Unternehmen kann Skaleneffekte nicht vollständig ausnutzen, denn verhältnismäßig zu hohe Produktionskosten stehen zu geringen Einnahmen gegenüber. Im anderen Extrem kann die Nachfrage nicht ausreichend befriedigt werden, man muss eventuell auf einen Teil der Gewinnmarge verzichten, verliert Marktanteile an Konkurrenten und somit hat man die Gelegenheit verpasst mehr Gewinn zu erwirtschaften. Das kann das Wachstum des Unternehmens bremsen, da durch eine geringere Profitabilität weniger Geld für Investitionen zu Verfügung steht. Diese beiden Beispiele sind salopp gesagt suboptimale Szenarien, die man als Unternehmer nach Möglichkeit verhindern möchte und sollte.

Die logische Konsequenz daraus ist sich auf mehrere Eventualitäten einzustellen und daraus eine Strategie abzuleiten, die diesen Erfordernissen gewachsen ist. Nachdem meistens mehr als zwei Zukunftsszenarien denkbar sind, gestaltet sich eine solche zukunftsorientierte Planung als nicht-triviale Herausforderung. Viele relevante Faktoren für die Planung sind nicht bekannt oder nur eingeschränkt prognostizierbar und Fehleinschätzungen führen zu den oben beschriebenen Szenarien. Bei der Lösung dieser Aufgabe werden mehrere Fragen aufgeworfen:

- Welche Szenarien sind denkbar?
- Wie passt man einen laufenden Plan aktuellen Ereignissen an?
- Kann man aus historischen Daten gewisse Schemata ableiten, um die Nachfrage besser einschätzen zu können?
- Woher bekommt man die entsprechenden Daten?
- Wie bestimmt man ein Produktionsoptimum, ohne die Zukunft zu kennen?

Diese Liste ließe sich sicher noch weiterführen und erhebt keinesfalls einen Anspruch auf Vollständigkeit. Die Problemstellungen im Kontext einer zukunftsorientierten Planung sollten damit allerdings mehr als deutlich dargelegt worden sein.

## **1.2. Ziel der Arbeit**

Die zentrale Problemstellung dieser Arbeit ist eine zukunftsorientierte Produktionsplanung. Zukunft heißt auch immer Unsicherheit, der es zu begegnen gilt. Dazu kommt in dieser Arbeit das mathematische Fachgebiet der Stochastik zur Anwendung, dass ja, wenn man einem Buchtitel Glauben schenken mag, Struktur in den Zufall bringen soll. Vgl. [1]. Das Mittel der Wahl sind filtrierte Wahrscheinlichkeitsräume. Da man die Realität nicht vollständig abbilden kann, muss man sich diverser Modelle bedienen, um in einer vereinfachten Abbildung der Realität zumindest eine fundierte Ahnung von einer zukünftigen Entwicklung haben zu können.

Dazu werden in dieser Arbeit die filtrierte Wahrscheinlichkeitsräume in Form eines Baumes als Datenstruktur modelliert. Dadurch werden die möglichen Entwicklungen vereinfacht ausgedrückt als Verzweigungen bzw. Äste im Baum dargestellt und geplant. Der Plan besteht also aus einer Summe von möglichen Endzuständen bzw. Blättern, die auf der Suche nach einer optimalen Lösung berücksichtigt werden. Die wertmäßigen Ausprägungen der möglichen Zustände innerhalb des Wahrscheinlichkeitsraumes werden dabei mit Hilfe eines additiven Random Walks, eines Vertreters aus der Familie der stochastischen Prozesse, bestimmt. Das hilft dem Anwender auch Alternativen zu entdecken, die vielleicht unberücksichtigt geblieben wären. Wenn der Plan ausgeführt wird, also bei der Produktionssteuerung, wird der Baum im Laufe der geplanten Perioden von der

Wurzel bis zu einem Blatt durchlaufen. Welche Äste dabei beschriftet werden und welches Blatt man im Endeffekt erreicht bleibt unbekannt, bis man den Planungshorizont erreicht hat. Im Laufe der Ausführung findet allerdings eine Informationsenthüllung statt und schafft damit die Möglichkeit, den Baum zu beschneiden und damit Äste und Blätter, die nicht mehr von Relevanz sind, auszuschließen und bei der Berechnung der optimalen Produktionsmenge nicht mehr zu berücksichtigen. Entscheidungskriterium hierbei ist die Umsatzergebnis-Funktion, welche namensgebend den aus der Absatzmenge erwirtschafteten Erfolg ermittelt. Die Zukunft verliert dadurch einen Teil ihrer Unsicherheit. Dabei kommen auch Prinzipien aus der Kybernetik zum Einsatz, welche einen Steuerkreis schaffen und den Plan einerseits flexibel, andererseits lernfähig machen.

Ziel dieser Arbeit ist es also, unter Verwendung der oben genannten Konzepte eine stochastisch optimale Produktionsplanung und –steuerung (SOPPS) als Modell zu konzipieren und in einem Prototyp implementiert umzusetzen. Der „Baum“ wird jedoch nicht auf einer „grünen Wiese“ gepflanzt, sondern soll sich nahtlos in einen Garten eingliedern, der sich ERP-Control nennt. Dieses Projekt unter der Federführung von Prof. Dr. Walter Schwaiger verfolgt das Ziel, alternative Konzepte aus der Ökonomie für die Anwendung in Business Performance Management Information Systems (BPM-IS) zu erforschen und in einer modernen IT-Architektur umzusetzen. Das Ergebnis soll eine neue Form von BPM-IS sein, welche mit alten Paradigmen bricht und zeigt, dass so manche Dogmen aus vergangenen Tagen überdacht werden sollten und es dazu sinnvolle Alternativen gibt.

### **1.3. Struktur der Arbeit**

Diese Arbeit gliedert sich im Wesentlichen in vier Schwerpunkte.

Kapitel 1 stellt das Thema der Arbeit vor und gibt Einblicke in die historische Entwicklung heutiger ERP-Systeme. Hier erfährt der Leser, aus welchem Umfeld die heutigen ERP-Systeme stammen und welche Überlegungen und Konzepte ihnen durch die gemeinsame Historie zu Grunde liegen. Dadurch soll ein Eindruck über das Metamodell der heutigen ERP-Systeme gewonnen werden.

In Kapitel 2 wird dann das Umfeld dieser Diplomarbeit näher beschrieben. Der Leser erhält Einblicke in das Projekt ERP-Control. In diesem Projekt wird ein ERP-System entwickelt, das sich im Metamodell von anderen ERP-Systemen unterscheidet. Der Schwerpunkt dieses Kapitels liegt im Metamodell von ERP-Control, also den Denkweisen und Methoden, die dabei zum Einsatz kommen und auch den in dieser Arbeit vorgestellten Prototyp als Teil von ERP-Control beeinflussen. Es werden die grundlegenden Konzepte erklärt und auch welche Standards und Normen bei der Umsetzung eines Prototyps berücksichtigt wurden und warum. In weiterer Folge wird auf das Thema dieser Arbeit eingegangen und welche Konzepte aus den unterschiedlichsten Forschungsgebieten der Mathematik, Wirtschaftswissenschaften und auch Informatik zur Anwendung kommen.

Das 3. Kapitel dokumentiert die Umsetzung des Prototyps einer stochastisch optimalen Produktionsplanung und –steuerung. Dabei wird auf die vorangegangenen Überlegungen im Detail eingegangen und argumentiert, warum welche Lösungswege gewählt wurden. Hier werden beispielhaft Auszüge aus dem Programmcode erörtert und die Verwendung der in dieser Arbeit vorgestellten Konzepte dokumentiert. Weiters wird die Benutzeroberfläche vorgestellt um dem Leser einen Eindruck vom Ergebnis der Arbeit zu verschaffen.

Den Abschluss bildet Kapitel 4, in dem die Arbeit kurz zusammengefasst wird und das Ergebnis der Arbeit sowohl im positiven wie auch im negativen Sinne diskutiert wird. Es werden mögliche Problemstellungen und Weiterentwicklungen für die Zukunft vorgeschlagen sowie ein vorsichtiger Blick in die zukünftige Entwicklung von ERP-Systemen gewagt.

## 1.4. Evolution von ERP

### MRP I:

Die Geschichte von Enterprise Resource Planning beginnen bereits in den 60er-Jahren des 19. Jahrhunderts. Bedingt durch die rasche Entwicklung der Computersysteme erkannten Unternehmen die Chance, ihr Tagesgeschäft durch den Einsatz von Computern zu beschleunigen und zu vereinfachen. Ab 1965 kann von den ersten Vorgängern von ERP-Systemen gesprochen werden. Material Requirements Planning-Systeme (MRP) waren, wie der Name schon erahnen lässt, Anwendungen zur Materialbedarfsplanung.

Bei Tom Burns werden MRP-Systeme wie folgt definiert:

„MRP is [...] a system which uses product definitions (bills-of-material), inventory and open order data, and master schedule information to calculate net, time-phased requirements for materials, components, and assemblies. It makes recommendations to release replenishment orders for inventoried items, and it reschedules open orders when due dates and need dates are not in phase . . . . it is a powerful technique for planning and controlling inventory as well as for scheduling production“[2]

Die Aufgaben, die MRP-Systeme zur damaligen Zeit ausführen konnten waren in der Produktionsplanung und Materialbedarfsplanung angesiedelt. Sie waren in der Lage ausgehend von der Absatzplanung aus dem Vertrieb einen Produktionsplan und in weiterer Folge auch die für die Produktion erforderliche Materialmengen zu ermitteln. Dabei wurden auch die Zeitpunkte berücksichtigt, zu denen die Produktion Arbeitsmittel benötigte, was Auswirkungen auf die Lagerhaltung hatte, welche nicht mehr mit kritischen Lagerständen bzw. Meldemengen sondern bedarfsorientiert agieren konnte. Weiters waren die Systeme in der Lage Termine für noch nicht eingeplante Aufträge vorzuschlagen. Die damals weit verbreiteten Mainframes, auf denen MRP-Systeme betrieben wurden, waren allerdings noch nicht in der Lage Datenverarbeitung in Echtzeit durchzuführen, weshalb sich das Konzept MRP auch auf Materialbedarfsplanung beschränken musste. Vgl. [3][4].

## **MRP II:**

Eine reine Materialbedarfsplanung war vielen Unternehmen mit wachsenden Kapazitäten der Computersysteme nicht mehr genug und so wurde aus MRP oder MRP I wie es in späterer Folge heißen sollte, MRP II. Diese Abkürzung stand nun für Manufacturing Requirements Planning.

Burns beschreibt MRP II als eine Kombination aus

„[...] purchasing, shop floor control, and capacity planning systems with basic MRP functionality. Purchasing systems determine when and what parts need to be ordered from vendors and let purchasing agents report supplier promised delivery dates. Shop floor control systems support dispatching and detailed scheduling control and allow line management to report current progress of open shop orders. The feedback obtained from these systems permits management to monitor and respond to actual conditions.“[2]

Neben der Materialbedarfsplanung wurden jetzt auch andere Unternehmensfunktionen in ein einheitliches System eingebunden. Exemplarisch seien hier nur Kapazitätsbedarfsplanung oder Absatz- und Vertriebsplanung erwähnt. Was an diesen Beispielen ersichtlich wird ist, dass bei MRP II nun alle für die Produktion notwendigen Ressourcen berücksichtigt werden. Die einheitliche Messgröße humaner, materieller und finanzieller Ressourcen wurde nun deren Realwert, also Geld. Damit hielt in MRP II eine finanzielle Perspektive Einzug, an der nun auch produktionsferne Bereiche eines Unternehmens anknüpfen konnten. Vgl. [3]. Nun wurden einzelne Abteilungen eines Unternehmens wie Lager oder Produktion nicht mehr getrennt betrachtet, sondern als Teile des Unternehmens, die miteinander interagieren. So rückten die Unternehmensprozesse immer weiter in den Fokus des Interesses und auch die logische Schlussfolgerung, sämtliche Unternehmensprozesse und alle Abteilungen eines Unternehmens in einem einheitlichen System abzubilden und zu steuern.

Das Aufkommen von relationalen Datenbanken, wie sie auch heute noch in ERP-Systemen Anwendung findet, sowie in den 1980er-Jahren die Verbreitung des Personal Computers (PC) begünstigten die vorhin beschriebene Entwicklungsansätze, da nun große Datenmengen effektiv gespeichert und bearbeitet werden konnten. Dank Tabellenkalkulation war nun auch eine

Datenanalyse auf dem PC möglich, ohne dafür einen Programmierer beschäftigen zu müssen. Vgl.[5].

### **ERP:**

Ab 1990 war diese Entwicklung soweit fortgeschritten, um echtes Enterprise Resource Planning (ERP) in Unternehmen umsetzen zu können. Ein wichtiger Schritt dorthin waren große Fortschritte in der Netzwerktechnik, die die Nutzung eines gemeinsamen Datenbestandes durch mehrere Anwender bzw. Abteilungen eines Unternehmens ermöglichte. Vgl. [5]. Nun war die Zeit reif für unternehmensweite Softwarelösungen, die nicht abgeschlossene Abteilungen sondern bereichsübergreifende Geschäftsprozesse (GP) abbilden konnten und so die Leitung eines Unternehmens auf eine ganz neue Art und Weise ermöglichten.

Es fand also ein Paradigmenwechsel weg von einer funktionsorientierten hin zu einer prozessorientierten Sichtweise statt. Es wurden in weiterer Folge nicht nur Unternehmensbereiche, die im weitesten Sinne etwas mit der Produktion zu tun haben eingebunden, sondern auch Abteilungen wie z.B. Personalmanagement, Finanz- und Rechnungswesen aber auch Forschung und Entwicklung oder inner- sowie außerbetriebliche Supportfunktionen. Als gemeinsamer Nenner zwischen den unterschiedlichen Bereichen eines Unternehmens und ihren Produkten und Dienstleistungen dienten wie bereits bei MRP II implementiert monetäre Messgrößen.

Dadurch wurde eine Bewertung des Beitrags der einzelnen Abteilungen am Wertschöpfungsprozess möglich, was dem Finanz- und Rechnungswesen sehr zu Gute kam. Anhand dieser Entwicklung ist es auch nicht verwunderlich, dass dieser Ansatz sich auch auf die Datenstrukturen hinter einer ERP-Software widerspiegelte. Die aus dem Rechnungswesen bekannten Konten repräsentierten nun sowohl Organisationsstrukturen innerhalb eines Unternehmens als auch Produkte und Dienstleistungen.

Martin Hesseler definiert ERP-Systeme in seinem Buch „Basiswissen ERP-Systeme“ folgendermaßen:

„Unter einem ERP-System wird allgemein eine integrierte betriebswirtschaftliche Standardsoftware verstanden. Mit ihr lassen sich betriebswirtschaftliche Aufgaben aus den verschiedenen Bereichen eines Unternehmens (z.B. Finanzwesen, Produktion, Logistik, Personalwesen) IT-gestützt bearbeiten.“[6]

In der jüngeren Vergangenheit gab es Versuche ERP II als Begriff für eine neue Generation von betrieblichen Informationssystemen zu etablieren. Das Beratungsunternehmen Gartner Inc.<sup>1</sup>, welches auch ERP als Begriff prägte, versuchte mit ERP II einen Begriff für die Symbiose zwischen innerbetrieblichen Informationssystemen und dem Schlagwort Web 2.0 zu finden.

#### **Web 2.0:**

Das Web 2.0 entwickelte sich evolutionär dank wachsender Rechnerleistungen sowohl bei Servern als auch PC aus dem Web 1.0 und wird durch die stärkere dynamische Ausrichtung von ebenjenem unterschieden. Das Web 1.0 hat zwar auch schon dynamische Aspekte, ist aber größtenteils statisch ausgelegt. Informationen wurden auf Internetseiten dargestellt und die Kommunikation fand mehrheitlich per E-Mail statt. Das Web 2.0 hat schon eine wesentlich dynamischere Ausrichtung zur Darstellung und Verbreitung von Informationen. Es lässt sich allerdings nicht scharf definieren, da es sich evolutionär entwickelte und sich nicht zu einem bestimmten Stichtag etablierte und eingesetzt wurde.

Klassische Beispiele für Web 2.0 sind Wikipedia, Twitter, Facebook oder YouTube. Inhalte werden von Anwendern für Anwender bereit gestellt und es sind keine spezialisierten Computerkenntnisse notwendig. Jeder kann mitmachen – die Kommunikation unter den Anwendern und deren Inhalt wird „lebendiger“. Dieses Merkmal war für Gartner wohl auch das entscheidende Unterscheidungsmerkmal zwischen ERP I und II. ERP II ist für Gartner eine Kommunikationsplattform innerhalb des Unternehmens und über dessen Grenzen hinaus.

---

<sup>1</sup> <http://www.gartner.com>



Tim O'Reilley hat 2007 versucht, die Kernkompetenzen von Unternehmen im Kontext von Web 2.0 zu identifizieren:

- ”
- services, not packaged software, with cost-effective scalability,
  - control over unique, hard-to-recreate data sources that get richer as more people use them,
  - trusting users as co-developers,
  - harnessing collective intelligence,
  - leveraging the long tail through customer self-service,
  - software above the level of a single device,
  - lightweight user interfaces, development models, AND business models.“[7]

Die Kernaussage ist auch hier, dass Produkte und Dienstleistungen im Web 2.0 im Vergleich zur Vergangenheit dynamischer ausgelegt sind. Der Trend geht weg von Softwarepaketen hin zu webbasierten Dienstleistungen. Auf diese Art wird auch der Informationsfluss dynamischer und nahezu jeder Mitarbeiter eines Unternehmens kann einen Beitrag zum Wissensschatz und damit auch zum Erfolg des jeweiligen Unternehmens beitragen. Gartner geht noch einen Schritt weiter und will den Informationsfluss nicht nur innerhalb eines Unternehmens vereinfachen sondern auch über die Unternehmensgrenzen hinweg mit Geschäftspartnern. Das erweitert das Konzept ERP um Supply Chain Management (SCM) und Customer Relationship Management (CRM) was eine Titulierung unter ERP II als neue Generation von ERP-Systemen durchaus rechtfertigen würde. Der Begriff setzte sich allerdings nicht durch und solche Komplettlösungen werden weiterhin unter dem Begriff ERP zusammengefasst.

## 2. Überblick über ERP-Control

Das folgende Kapitel gibt einen kurzen Überblick über das Projekt ERP-Control, welches unter der Federführung von Univ.Prof. Mag. Dr. Walter Schwaiger am Institut für Managementwissenschaften der TU Wien durchgeführt wird. Ziel des Projektes ist es den Prototyp eines ERP-Systems zu entwickeln und dabei neue Wege beim theoretischen Unterbau zu gehen. Die dem System zugrundeliegenden Konzepte werden in weiterer Folge im Abschnitt zum Thema Metamodell näher erläutert und es werden auch die Unterschiede zu klassischen heute im Einsatz befindlichen ERP-Systemen diskutiert.

### 2.1. Architektur eines ERP-Systems

#### Exkurs Systemarchitektur:

Heutzutage gebräuchliche webbasierte Systeme, und damit sind in diesem Fall nicht ausschließlich nur ERP-Systeme gemeint, basieren auf einer mehrschichtigen Systemarchitektur. Die Gründe für die Auswahl dieses Strukturierungsprinzips bei der Implementierung von Softwaresystemen sind mannigfaltig. Ein Hauptgrund ist das Bestreben, die Abhängigkeit der Systemkomponenten untereinander zu minimieren. Das vereinfacht die Entwicklung sowie auch die Wartung der Applikation und in weiterer Folge auch deren Weiterentwicklung und Erweiterung. Damit wird auch der Datenbestand von der Logik getrennt, was wiederum die Wartung des Datenbestandes wesentlich vereinfacht. Ein weiterer Vorteil ist eine erhöhte Sicherheit, die sich aus der sauberen Trennung der einzelnen Komponenten eines Systems sowie deren Schnittstellen untereinander ergibt. Zu den weitverbreitetsten Modellen zählt das in Abbildung 1 illustrierte 3-Schichten-Modell.

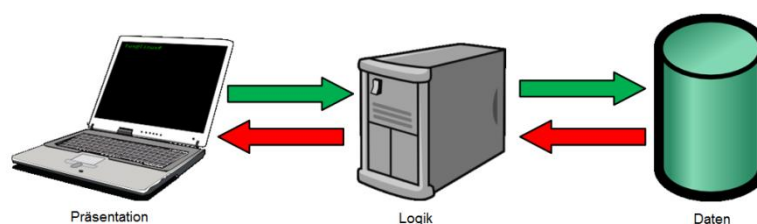


Abbildung 1: Drei-Schichten-Architektur

Dieses Architekturmodell unterscheidet Datenerhaltungsschicht, Logikschicht und Präsentationsschicht. Ganz rechts in der Abbildung sieht man die Datenbank, welche sich in der Datenerhaltungsschicht befindet. Diese enthält Geschäftsdaten, auf denen die Logik operiert. Die Logik zur Datenverwaltung wird ebenfalls in dieser Schicht gekapselt und ist meist eine eigenständige Software. In der Mitte wird mit dem Application Server die Logikschicht repräsentiert. Auf diesem Server läuft die Anwendungslogik, also die eigentlichen Programme. Diese Schicht hat Zugriff auf die Datenerhaltungsschicht und kann die dort gespeicherten Daten manipulieren. Ganz links befindet sich die Präsentationsschicht. Sie ist nur für die Darstellung von Daten bzw. daraus berechneter Ergebnisse zuständig und kann auch nur mit der Logikschicht interagieren.

### **Exkurs Programmarchitektur:**

Ein weiteres Prinzip, welches ebenfalls bei der Softwareentwicklung einen breiten Zuspruch und Anwendung findet, ist eine Serviceorientierte Architektur (SOA). Auch dieser Begriff wurde erstmalig von Gartner Inc. geprägt. Wie bereits im Abschnitt über die Evolution von ERP-Software erwähnt, ging man bei ERP von einer funktionsorientierten Sichtweise über zu einer prozessorientierten Sicht auf ein Unternehmen. Genau diese Philosophie wird auch bei SOA verfolgt. Ein Programm soll nicht mehr nur Funktionalitäten erfüllen, sondern vielmehr eine Reihe von Services zu Verfügung stellen. Sinn dahinter ist es einzelne Funktionalitäten einer Applikation zu kapseln und diese nach dem Baukastenprinzip dazu zu verwenden um GP informationstechnisch abzubilden. In diesem Zusammenhang spricht man auch von Orchestrierung, da einzelne Services zu einem Prozess verbunden werden, anstatt jede Funktionalität in einem GP einzeln implementieren zu müssen. Christoph Mathas definiert SOA in seinem Buch „SOA intern“ folgendermaßen:

„In einer Service-orientierten Architektur handelt es sich um ein generisches Entwurfsmuster für IT-Systemarchitekturen oder [...] Systemarchitektur–„Pattern“. SOA beschreibt eine Art und Weise, ein lose gekoppeltes, verteiltes Software-System zu entwerfen und zu realisieren, sowie seine Einzelbausteine (vor allem Services) untereinander und auf verschiedenen Abstraktions- und Skalierungsebenen in eine sinnvolle Wechselwirkung zu setzen.“[8]

## 2.2. ERP-Control

Um dem Leser einen Eindruck von ERP-Control zu vermitteln wird die Software im Folgenden kurz vorgestellt. Im derzeitigen Entwicklungsstand sind etliche Funktionalitäten vorgesehen, welcher in Abbildung 2, der Startseite der Applikation, aufgelistet sind. Durch eine prozessorientierte Benutzeroberfläche wird der Anwender Schritt für Schritt durch einen GP geführt. Das verhindert, dass einzelne Aufgaben innerhalb eines Prozesses vergessen oder in der falschen Reihenfolge durchgeführt werden.

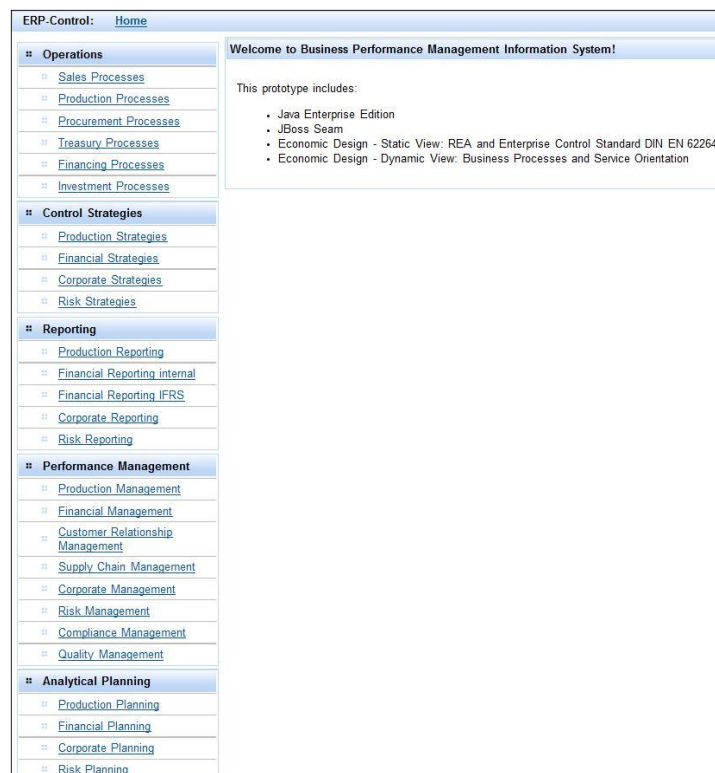
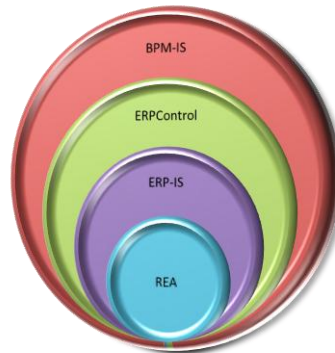


Abbildung 2: ERP-Control

Im Operations-Bereich werden die häufigsten GP des Tagesgeschäfts zusammengefasst. Das umfasst unter anderem Funktionalitäten für die Durchführung von Verkauf, Produktion, Beschaffung oder Finanzierung und Investment. Der Bereich Control Strategies beschäftigt sich mit der Steuerung von Produktion, Finanzanlagen oder auch des Risikos. Reporting erfüllt selbstredend Funktionalitäten zur Berichterstattung. Das Performance Management ist befasst mit Effizienzmessungen in den bisher schon aufgezählten Bereichen und geht damit über die konventionellen Anwendungsgebiete von ERP-Software hinaus hin zu Business Performance

Management. Analytical Planning beinhaltet Funktionen zur Analyse und Simulation von Plänen aus den unterschiedlichen Unternehmensbereichen.

Abbildung 3 zeigt dabei den konzeptionellen Aufbau hinter ERP-Control. Die nachfolgend erwähnten Konzepte, Standards und Normen werden in weiterer Folge dieser Arbeit noch näher erläutert.



**Abbildung 3: ERP-Control Konzept**

Den Kern bildet die REA-Ontologie (siehe Abschnitt 2.4.1), welche erweitert durch den AEO-Standard (Abschnitt 2.4.3.2) die Grundlage für die Betrachtung und Abbildung der relevanten Daten eines GP in der Datenstruktur darstellen. Darauf aufbauend wurde ein ERP-Information System (ERP-IS) aufgesetzt, das ebenfalls beeinflusst durch den AEO-Standard eine Ressourcenbasierte Planung realisiert. Dieses System wird durch den ECSI-Standard (siehe Abschnitt 2.4.3.1), welche die Beschreibung geschäftsrelevanter Daten normiert, sowie den Controlling-Konzepten Steuerung und Regelung (siehe Abschnitt 2.4.2) erweitert. Das Resultat ist ein System, welches Performance Management ermöglicht, also die Steuerung und Messung der Leistung innerhalb eines Unternehmens ermöglicht. Die letzte Ausbaustufe stellt das Business Performance Management Information System (BPM-IS) dar, welches unterstützt von den darunter liegenden Konzepten eine unternehmensweite Betrachtung ermöglicht und so Entscheidungsgrundlagen und Planungsfunktionalitäten für die Unternehmensführung bereit stellt. Dabei wird der konventionelle Ansatz eines Management Information Systems (MIS) abermals durch Controlling-Konzepte erweitert.

Das Resultat ist ein System, welches nicht nur klassische ERP- und MIS-Funktionalitäten abdeckt, sondern aufgrund neuer Konzepte eine intuitive Sichtweise auf die GP eines Unternehmens ermöglicht und dabei auch

kybernetische Konzepte im Controlling aufgreift, um diese zu regeln und steuern. Durch die REA-Ontologie wird ein GP tatsächlich als Prozess gesehen und verarbeitet anstatt buchhalterisch als Buchungssätze zu betrachten. Die Einbindung etablierter Standards erleichtert das Verständnis der zugrundeliegenden Datenstrukturen und ermöglicht eine Einbindung in bestehende Systemarchitekturen eines Unternehmens. Die Verwendung von Controlling-Konzepten aus der Kybernetik schafft bei der Planung, Steuerung, Kontrolle und Ausführung der GP und unternehmensweit der Geschäftstätigkeit an sich einen leicht verständlichen Regelkreis, der Sicherheit schafft und Risiken reduziert.

### **2.3. Abgrenzung zu anderen MIS-Systemen**

Wie bereits in der Geschichte von ERP angeschnitten wurde handelte es sich bei ERP-Systemen nicht um eine Neuentwicklung, sondern es fand eine Weiterentwicklung auf Basis bestehender Konzepte statt, um neuen Anforderungen gerecht zu werden und neue technische Möglichkeiten nutzen zu können. Genau dieser Umstand führt auch zu einem wesentlichen Nachteil von etablierten ERP-Systemen. Sie entwickelten sich evolutionär aus früheren Versionen und wurden ständig erweitert, aktualisiert und um neue Funktionalitäten ergänzt. Die Kernfunktionalitäten sind jedoch unverändert geblieben und damit auch die Konzepte – das Metamodell – auf denen sie aufsetzen.

Neuen Anforderungen wurden in der Programmlogik durch Adaptionen bestehender Funktionalitäten oder neue Programmmodule Rechnung getragen. Dieser Umstand hat zur Folge, dass die Architektur des Datenmodells unter Umständen nicht sauber von der Applikationslogik getrennt werden kann. Das Architekturmodell SOA wurde erst 1996 von Gartner erstmals erwähnt und kam daher erst nach der Entwicklung der ersten ERP-Systeme zur Anwendung. Ein Grund dafür diese unscharfe Trennung mag sein, dass neue Anforderungen auch unter Umständen neue Datentypen implizieren. Diese werden jedoch bei einer Weiterentwicklung oder Adaption eines bestehenden Systems von aus bestehenden Datentypen abgeleitet.

Als Beispiel seien hier Finanzprodukte erwähnt, die für den Produktionssektor und dem damaligen MRP-Systemen, aus dem sich ERP ja entwickelte, eine untergeordnete bis gar keine Rolle spielten. Bei einer integrierten Gesamtsicht auf ein Unternehmen werden Finanzprodukte relevant, unterscheiden sich jedoch von materiellen Produkten oder Dienstleistungen durch diverse Attribute wie z.B. ein finanzielles Risiko oder eine Bewertung des Finanzproduktes, welche auch spezielle Funktionalitäten erfordern. Nachdem man eine Neuentwicklung häufig wesentlich kostspieliger ist als eine Adaption, vor allem bei so umfangreichen Applikationen wie einer ERP-Software, wird eben der Kern des Programms unverändert gelassen und darauf aufbauend werden die Neuerungen hinzugefügt.

Diesem Umstand wird in ERP-Control Rechnung getragen. Durch Konformität mit dem ECSI-Standard wird sichergestellt, dass sich eine ERP-Software als Schnittstelle zwischen Produktion und dem Management anhand des Standards in ein Unternehmen einbinden lässt. Durch die Definition von generisch gehaltenen Datenmodellen, welche im Standard beschrieben werden, kann flexibel auf sich ändernde Anforderungen reagiert werden ohne auf ältere Konzepte zurückgreifen zu müssen.

Ein weiteres Beispiel ist die prozessorientierte Sichtweise auf ein Unternehmen. Nachdem diese Art der Betrachtung eines Unternehmens ebenfalls erst später Verbreitung fand, findet sich in etablierten ERP-Lösungen keine explizite Repräsentation eines GP. Die Verknüpfung zwischen Geschäftsdaten und einzelnen Ereignissen innerhalb eines GP wird durch die Applikationslogik hergestellt. Bei ERP-Control wird dies schon in der Datenstruktur berücksichtigt, welche durch die REA-Ontologie Geschäftsdaten und Ereignisse schon dort als Prozess abbildet. Nachdem ERP-Control eine Neuentwicklung darstellt, konnte schon bei der Planung des Systems der aktuelle Stand der Wissenschaft sowie aktuelle Anforderungen an ein MIS berücksichtigt werden.

## **2.4. Metamodell von ERP-Control**

Nachfolgend werden jene Konzepte vorgestellt, die für ERP-Control das Metamodell, also die theoretische Grundlage bilden. Allen Voran und deshalb auch zu aller Erst vorgestellt wird die REA-Ontologie, mit der Geschäftsprozesse nicht durch buchhalterische Mittel dargestellt wird, sondern wesentlich intuitiver als tatsächlichen Prozess. Im Anschluss daran werden kybernetische Lenkungsprinzipien erläutert, welche ebenfalls einen konzeptionellen Grundstein für ERP-Control darstellen. Weiters werden die verwendeten Normen und Standards in Auszügen vorgestellt und welchen Nutzen sie für das Projekt haben. Den Abschluss dieses Abschnittes bilden Begriffsdefinitionen einiger Konzepte, welche für diese Arbeit ebenfalls von Bedeutung sind.

### **2.4.1. REA-Ontologie**

Obwohl REA in den nachfolgenden Normen ebenfalls Anwendung findet, sei diesem Thema ein eigener Abschnitt gewidmet, da es ein zentrales Konzept von ERP-Control darstellt und deshalb von besonderer Bedeutung ist. Die REA-Ontologie geht zurück auf William E. McCarthy. Er hat sich während seiner akademischen Laufbahn besonders in Rechnungswesen in Verbindung mit Informatik hervorgetan. Derzeit hat er eine Professur für Buchhaltung an der Michigan State University inne und ist in mehreren Gremien wie z.B. UN/CEFACT oder ISO vertreten.

Eines seiner bekanntesten Arbeiten ist die REA-Ontologie. Diese beinhaltet im Wesentlichen eine neue Sicht auf Geschäftsfälle, welche üblicherweise in der Buchhaltung durch Buchungssätze abgebildet werden. Monetäre Werte wechseln hier von einem Konto auf das andere. Der dazugehörige Geschäftsfall an sich wird dabei nur unbefriedigend bis gar nicht abgebildet. McCarthy war die kontenbezogene Sichtweise auf einen Geschäftsfall sowie die damals verwendeten Datenmodelle nicht aussagekräftig genug. Aus diesem Grund wollte er einen Geschäftsfall auf eine intuitive Art und Weise in einer relationalen Datenbank abbilden können um sowohl buchhalterischen Anforderungen gerecht zu werden als auch informationstechnische Konzepte so gut wie möglich auszunutzen. Er strebte nach einer Unterscheidung zwischen Ressourcen und Agenten verbunden durch einen Event, welcher den Geschäftsfall symbolisiert.



Bei den Ressourcen handelt es sich um Dienstleistungen, materielle oder finanzielle Güter während ein Agent eine agierende Einheit in einem Unternehmen wie z.B. eine Person, eine Abteilung udgl. darstellt. Nachfolgend wird das Konzept von REA näher erläutert.

Die REA-Ontologie wurde erstmals im Juli 1982 von McCarthy in „The Accounting Review“ publiziert. Das Akronym REA steht dabei namensgebend für Ressourcen, Events und Agenten. Mit Hilfe dieser Klassifizierung ist es leicht möglich einen Geschäftsfall in eine relationale Datenbank abzuspeichern. Folgende Abbildung 4 [9] zeigt das Prinzip von REA.

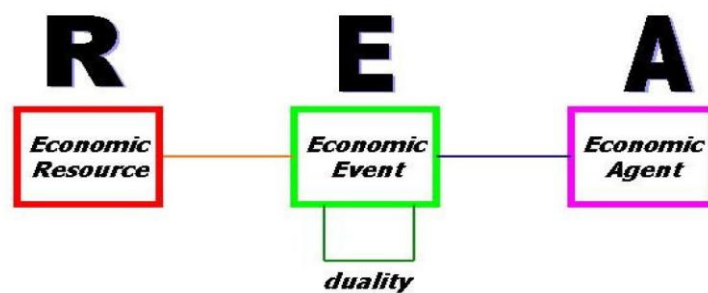


Abbildung 4: REA-Ontologie

Eine REA-Entität besteht wie schon beschrieben aus Ressourcen, Ereignisse und Agenten. In einem Ereignis treten also Veränderungen von Ressourcen ein, die von Agenten angestoßen werden. Ein wichtiges Prinzip dabei ist die Dualität. Diese ist vergleichbar mit einer Buchung und der entsprechenden Gegenbuchung. Beispielsweise bei einem Verkauf wechselt ein Produkt von Verkäufer zum Käufer und auf dem umgekehrten Weg erhält der Verkäufer vom Käufer Geld für das Produkt. Zwei Ressourcen, nämlich Geld und ein Produkt wurden dabei im Ereignis Verkauf manipuliert, es kommt zu einem Ressourcenaustausch.

Die obige Abbildung ist sehr generisch gehalten und soll nur das Prinzip und die verwendeten Klassen von unterschiedlichen Objekten in der REA-Ontologie veranschaulichen. Tatsächlich sind an einem Ereignis meist mehr als ein Agent und mehr als eine Ressource betroffen. Dies wird allerdings nur durch das Dualitätsprinzip angedeutet, welches sich auf den Austausch von Ressourcen bezieht. Eine einseitige Transaktion ohne Gegenleistung wird dadurch ausgeschlossen.

## 2.4.2. Kybernetische Betrachtung eines Systems

Nachfolgend wird in die Wissenschaft der Kybernetik und hier im Speziellen in das kybernetische Steuerungsprinzip kurz eingeführt. Die Kybernetik als Wissenschaft beschäftigt sich im Wesentlichen mit der Steuerung und Regelung von Systemen. Abgesehen davon bietet diese Wissenschaft eine abstraktere Sichtweise auf Konzepte wie Information oder Kommunikation, welche ja die Vehikel für die Manipulation bzw. Lenkung eines Systems darstellen. Der Begriff Kybernetik lässt sich bis in das Jahr 1947 zurückverfolgen und wurde von Dr. Norbert Wiener geprägt, welcher Kybernetik wie folgt definiert:

„Wir haben beschlossen, das ganze Gebiet der Regelung und Nachrichtentheorie, ob in der Maschine oder im Tier, mit dem Namen >>Kybernetik<< zu benennen, den wir aus dem griechischen >>κυβερνήτης<< oder >>Steuermann<< bildeten. Durch die Wahl dieses Ausdrucks möchten wir anerkennen, dass die erste bedeutende Schrift über Rückkopplungsmechanismen, ein Artikel über Fliehkraftregler von Clerk Maxwell, im Jahre 1868 veröffentlicht wurde, und dieses englische Wort >>Governor<< für Fliehkraftregler ist von einer lateinischen Verfälschung von κυβερνήτης abgeleitet. Wir wollen auf die Tatsache verweisen, dass die Steuermaschine eines Schiffes tatsächlich eine der ersten und am besten entwickelten Formen von Rückkopplungsmechanismen ist.“ [10]

Im Laufe ihrer Forschung hatten sich Wissenschaftler rund um Dr. Arturo Rosenblueth und Dr. Norbert Wiener die Gesetzmäßigkeiten von Steuerung als Studienobjekt gewählt. Dabei wählten sie einen generischen Ansatz und unterschieden dabei nicht zwischen der Steuerung von Maschinen, Tieren oder Menschen. Durch diese Sichtweise erlangten sie ein neues Verständnis für die Begriffe Information, Steuerung und Rückkopplung – drei der wesentlichsten Elemente in der Kybernetik. Sie hatten gemeinsame Problemstellungen in den Forschungsgebieten „Nachrichtenübertragung, Regelung und Statistische Mechanik erkannt, sowohl bei der Maschine als auch im lebenden Gewebe.“ [10]

Der konventionelle Ansatz Fragestellungen unterschiedlicher Fachgebiete getrennt zu betrachten bringt den Umstand mit sich, dass sich diese Fächer im Laufe der Zeit immer weiter spezialisieren und damit nicht nur Überlappungen

untereinander aufweisen, sondern als Konsequenz auch manche Themen unbehandelt lassen, die nur im weitesten Sinne dem jeweiligen Fachgebiet zugeordnet werden können. Nachdem für eine solche Art von fächerübergreifenden Problemen in der Fachliteratur keine eindeutige Terminologie zu finden war, waren sie gezwungen,

„wie es Wissenschaftlern so oft ergeht, [...] einen künstlichen neogriechischen Ausdruck zu prägen, um die Lücke zu füllen“ [10]

und einen Terminus für die Wissenschaft von Steuerungs- und Regelvorgängen zu schaffen und das „interdisziplinäre Niemandsland“, wie es Rolf Rothmayer Vgl. [11] bezeichnete, zu beseitigen. So entstand 1947 die Kybernetik als Wissenschaft, die auch umgangssprachlich als Kunst des Steuerns bezeichnet wird.

Für diese Arbeit besonders interessant sind dabei die Konzepte der Regelung und Steuerung, welche nachfolgend kurz vorgestellt werden.

#### **Kybernetisches Regelungsprinzip:**

Beim kybernetischen Regelungsprinzip findet eine Rückkopplung statt. Hier hat sich auch der englische Begriff Feedback in der Alltagssprache etabliert. Anhand einer vordefinierten Stellgröße wird laufend der Zustand des zu regelnden Systems korrigiert. Es findet also ein Soll-Ist-Vergleich statt. Um bei Wieners Seefahrtanalogie zu verbleiben stelle man sich vor, anstatt eines Steuermanns übernimmt ein Autopilot das Ruder. Der Autopilot folgt einem vorgegebenen Kurs und misst dabei, ob der Kurs vom vorgegebenen Kurs abweicht. Ist das der Fall, so ergreift der Autopilot Gegenmaßnahmen und korrigiert den Kurs hin zur vorgegebenen Kurs. Dieses Konzept bezeichnet man auch als Selbststeuerung des Systems. Ein Eingriff von außen ist hier nicht notwendig. Der Nachteil an einer Feedback-Regelung ist der vergangenheitsbezogene Grundgedanke. Erst nach dem Auftreten einer Abweichung vom Soll-Wert kann reagiert werden.

### **Kybernetisches Steuerungsprinzip:**

Im Gegensatz dazu verfolgt die Steuerung ein anderes Konzept, jenes des Feedforward oder Vorkopplung. Der französische Kybernetiker Albert Ducrocq beschrieb das Konzept der Steuerung folgendermaßen:

„Ein System beherrschen heißt tatsächlich, ihm in einem bestimmten Augenblick der Zukunft einen wohldefinierten Zustand zuzuweisen und dabei alle anderen möglichen Zustände auszuschließen. In diesem Sinnen wollen wir den Begriff der Steuerung vor allem auffassen als eine Neutralisierung vom Zufall abhängiger Handlungen, also als einen Kampf gegen den Zufall“  
[12]

Beim kybernetischen Steuerungsprinzip erfolgt die Manipulation des Systems durch eine nicht systeminhärente Größe, dem Steuermann sozusagen. Diese Steuerung erfolgt nicht durch einen retrospektiven Soll-Ist-Vergleich, was das Zufallsmoment für zukünftige Zustände ja nicht eliminieren würde und der Rückkopplung aus der Regelung entsprechen würde, sondern hier kommt eine zukunftsorientierte Sichtweise zu tragen. Anstatt sich an einem Soll-Wert zu orientieren, wird nach Ducrocq ein Plan erstellt, welcher die möglichen zukünftigen Zustände eines Systems vorweg nimmt und so die Zukunft in gewisser Weise berechenbar macht. Dazu werden mögliche Störgrößen bei der Planung berücksichtigt und die Stellgröße für die Zukunft dementsprechend berechnet, um eine zufriedenstellende Realisation des Planes möglichst sicher zu stellen. Im Unterschied zur Regelung ist man hier nicht auf den ständigen Input von Ist-Werten angewiesen.

Der gravierende Nachteil dieser Methode ist jedoch die zuverlässige Planung und Vorhersage der Störgrößen. Sind die zugrundeliegenden Prognosen falsch und wird so die Realität unzureichend im Plan abgebildet, dann sind auch fundierte und weit entwickelte Planungstechniken machtlos.

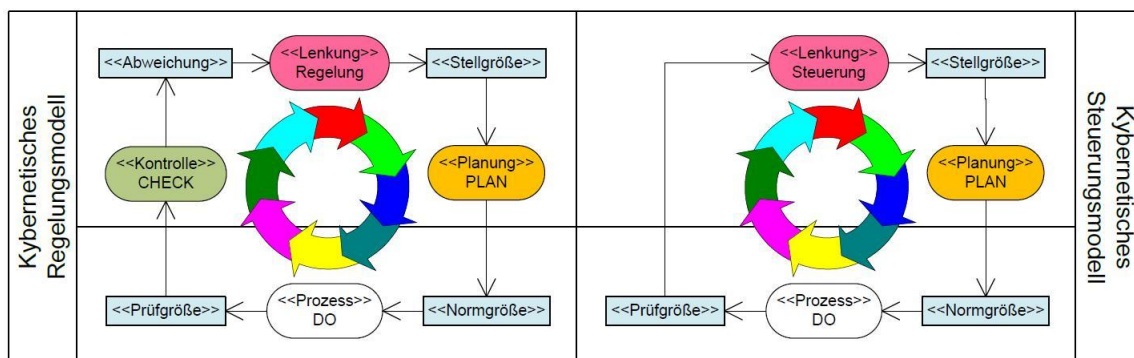
### **Exkurs PDCA-Zyklus:**

Im Zusammenhang mit Kybernetik darf auch der populärere PDCA-Zyklus nicht unerwähnt bleiben. Dieses Konzept kommt ursprünglich aus dem Qualitätsmanagement, hat sich aber im Laufe der Zeit auch in anderen Bereichen etabliert. Das Konzept ersann der amerikanische Physiker Walter. A. Shewhart und wurde von seinem ebenfalls aus den USA stammenden Fachkollegen

William Edwards Deming 1950 in Japan zitiert und publik gemacht. Deming benannte die Idee auch Shewhart cycle, sie wurde aber in Japan dennoch unter Deming cycle bekannt und hat sich bis heute hartnäckig gehalten. Wegen der vier Schritte (Plan, Do, Check, Act) des Zyklus ist auch die Bezeichnung PDCA-Zyklus gebräuchlich. Ein ähnliches Konzept wird auch bei KVP (kontinuierlicher Verbesserungsprozess) oder dem japanischen Pendant Kaizen angewandt.

Wie bereits erwähnt enthält der PDCA-Zyklus vier Schritte, welche iterativ durchgeführt werden. Schritt 1, **PLAN**, beschäftigt sich selbstredend mit der Planung eines Geschäftsprozesses. In diesem Schritt wird die Ist-Situation analysiert sowie mögliche Verbesserungsvorschläge fließen in die Planung ein. In Schritt 2, **DO**, wird der zuvor erarbeitete Plan ausgeführt und die Ergebnisse und Prozessparameter werden für eine spätere Überprüfung protokolliert. Anschließend folgt die Kontrolle oder **CHECK**. In diesem Schritt wird anhand einer Zielgröße der Erfolg der Durchführung gemessen und allfällige Abweichungen dadurch festgestellt. Die Iteration des Zyklus findet im Schritt **ACT** ihr Ende. Hier wird auf die gewonnenen Erkenntnisse reagiert und der Plan für die nächste Iteration adaptiert.

Abbildung 5 [13] illustriert den PDCA-Zyklus (links) als kybernetisches Regelungsmodell und stellt es der kybernetischen Steuerung gegenüber.



**Abbildung 5: Kybernetische Managementmodelle - Unterscheidung von Regelungs- und Steuerungsmodell**

Diese Abbildung verdeutlicht auch den gravierenden Unterschied zwischen kybernetischer Regelung und Steuerung. Während die Regelung aus der Vergangenheit lernt und dabei den Plan durch Eliminierung der entstandenen Abweichung korrigiert, kommt die Steuerung ohne vergangenheitsbezogene Werte aus, da Abweichungen oder Störgrößen bereits in der Planung berücksichtigt wurden.

### 2.4.3. Relevante Normen

Das Austrian Standards Institute definiert eine Norm folgendermaßen:

„Eine Norm ist eine qualifizierte Empfehlung - kein Gesetz -, die im Konsens nach international anerkannten Verfahren erstellt wurde. Sie beruht auf abgestimmten Ergebnissen von Wissenschaft, Technik und Praxis und zielt auf größtmöglichen Nutzen für alle. Sie wird von einer anerkannten Normungsorganisation zur allgemeinen und wiederkehrenden Anwendung angenommen und ist öffentlich zugänglich.“ [14]

Eine Norm stellt also eine Grundlage von Zusammenarbeit dar, indem sie international anerkannte Rahmenbedingungen und Regeln definiert. Eine Normierung hilft auch Geschäftspartnern bei der Einschätzung von Leistungen und Produkten. Dadurch kann unter anderem ermittelt werden, ob und in wie weit ein angebotenes Produkt für einen Kunden geeignet ist und zu welcher Qualität dieses angeboten wird. Desweiteren enthält eine Norm auch Vorschläge zu deren Umsetzung oder weist auf Schwierigkeiten und Probleme hin, sodass man aus den früheren Fehlern anderer, welche bei der Erstellung der Norm mitgewirkt haben, lernen kann. Eine gemeinsame Wissensgrundlage hilft auch dabei, die Arbeit eines Anderen besser und vor allem schneller zu verstehen. Beispielsweise wurden die Literaturverweise dieser Arbeit nach der ISO 690-Norm gekennzeichnet, damit der Leser sich durch ein standardisiertes und vertrautes Layout leichter zurecht findet.

Man sieht also, dass eine Norm aus einer Vielzahl von Gründen sinnvoll ist und nach Möglichkeit auch Anwendung finden sollte. Aus diesem Grund wurden auch in ERP-Control Normen und Standards berücksichtigt. Der nachfolgende Abschnitt stellt die in dem Projekt verwendeten Normen und Standards vor und geht dabei detailliert auf die für ERP-Control besonders relevanten Teile ein, da manche Normen und Standards ursprünglich nicht für die Verwendung in ERP-Systemen vorgesehen waren und sind, dessen Berücksichtigung aber durchaus sinnvoll erscheint.

### **2.4.3.1. ECSI-Standard (DIN EN 62264)**

Der DIN EN-Standard 62264 trägt den Namen „Integration von Unternehmens und Leitsystemen“ und ist in seiner aktuellen Form gültig seit Juni 2008. Ihm vorangegangen ist der in den USA entstandene ANSI/ISA 95, welcher im europäischen Raum als DIN EN 62264 verbreitet wurde. Er wurde Enterprise-Control System Integration (ECSI) benannt, hat aber im Wesentlichen denselben Inhalt. Beide Standards definieren

„Schnittstellen zwischen Unternehmensaktivitäten und Aktivitäten der leittechnischen Domäne“ und beinhalten „Normenmodelle zur Beschreibung der Schnittstellen zwischen den Geschäftssystemen eines Unternehmens und seinen Produktionssteuerungssystemen.“

[15]

Der Fokus dieser Standards liegt in der fachgerechten Zusammenführung von Managementsystem und den Produktivsystemen eines Unternehmens. Dabei sollen Synergieeffekte aus beiden Bereichen eines Unternehmens genutzt werden. Die Standards stellen dazu unterschiedliche Modelle und Konzepte zur Realisierung dieser Integration von Management- und Produktivsystemen vor, welche vom Automatisierungsgrad her unabhängig sein sollen. Klare Ziele der Standards und deshalb auch in deren Einleitungen erwähnt sind:

- Verkürzung des benötigten Zeitraums bis zur vollen Produktion von neuen Produkten
- Hersteller von einschlägiger Software sollen in die Lage versetzt werden, entsprechende Werkzeuge zur oben beschriebenen Integration anzubieten
- Der Anwender soll durch Anwendung der Standards seine Anforderungen besser definieren können.
- Kosten für die Automatisierung von Prozessen sollen reduziert werden.
- Supply Chains sollen unternehmensübergreifend optimiert werden. Vgl. [15]

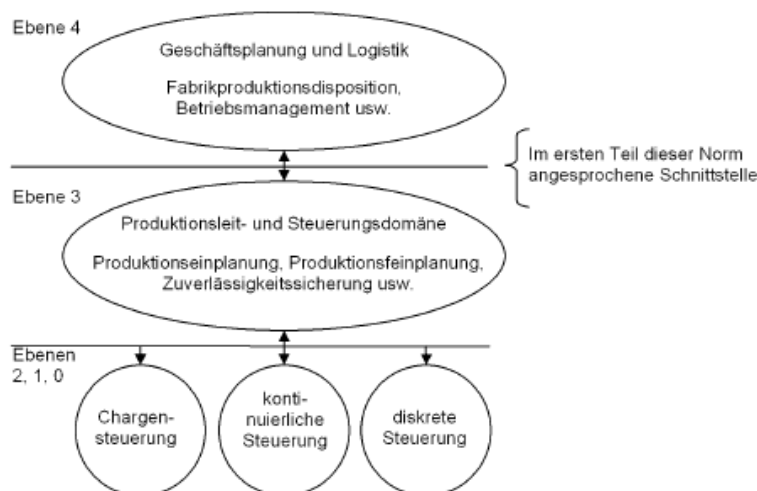
Der ECSI-Standard besteht aus drei Teilen, welche nachfolgend kurz vorgestellt werden. Der Standard fokussiert dabei auf Produktionsunternehmen, weshalb die

darin beschriebenen Konzepte sich auch sehr stark an den Schnittstellen zwischen Management und Produktion orientieren.

Der erste Teil beschäftigt sich mit Modellen und Terminologien um die oben beschriebenen Ziele erreichen zu können. Zu diesen Modellen zählen Hierarchiemodelle zur Beschreibung der leittechnischen Domänen, ein Datenflussmodell zur Beschreibung der funktionalen Beziehungen sowie dem damit verbundenen Datenaustausch zwischen Organisationseinheiten innerhalb des Unternehmens sowie ein Objektmodell, welches die benötigten Informationen für eben jenen Datenaustausch definiert.

Im Abschnitt über Hierarchiemodelle werden zwei Konzepte vorgestellt. In einer Anlagenhierarchie werden physische Ausprägungen eines Unternehmens und deren hierarchischer Zusammenhang erläutert. Darunter versteht man beispielsweise, dass ein Unternehmen aus mehreren Betriebsstätten besteht, welche sich aus Anlagenkomplexe zusammensetzen. Diese werden dann wiederum in Arbeitsbereiche, Produktionsteilanlagen oder Produktionslinien unterteilt und für jede dieser Ausprägungen werden definiert.

Für das Projekt ERP-Control wesentlich interessanter ist das funktionale Hierarchiemodell in Anlehnung an das Purdue Referenzmodell für CIM, welches sich ebenfalls im Anhang des Standards finden lässt.



**Abbildung 6: Funktionale Hierarchie**

Hier wird auf vier Ebenen die Entscheidungshierarchie innerhalb eines Unternehmens abgebildet. Wie in der Abbildung 6 [15] bereits angemerkt wird, behandelt der erste Teil des Standards eben die Schnittstelle zwischen



Managementfunktionen und der oberen Ebene der Betriebsleitung. Genau hier liegt auch das Betätigungsfeld von ERP-Control. Weiters werden die typischen Aktivitäten der einzelnen Ebenen beschrieben. In Ebene 4 finden sich unter Anderem auch die Erstellung und Anpassung von Produktionsplänen wieder – das Thema dieser Arbeit. Auf Ebene 3 findet sich das operative Management wieder, das sich mit dem Tagesgeschäft, in Produktivunternehmen also mit der Produktion, beschäftigt. Die Ebenen 2 bis 0 befassen sich mit der Durchführung der GP und dessen Controlling.

Ein weiteres für ERP-Control wichtiges Konzept ist das Funktionale Unternehmensführungs- und Leitsystemmodell. In diesem Modell werden die einzelnen Funktionen des Unternehmens und deren Schnittstellen untereinander in Beziehung gebracht. In diesem Modell werden die einzelnen Funktionen des Unternehmens und deren Schnittstellen untereinander in Beziehung gebracht.

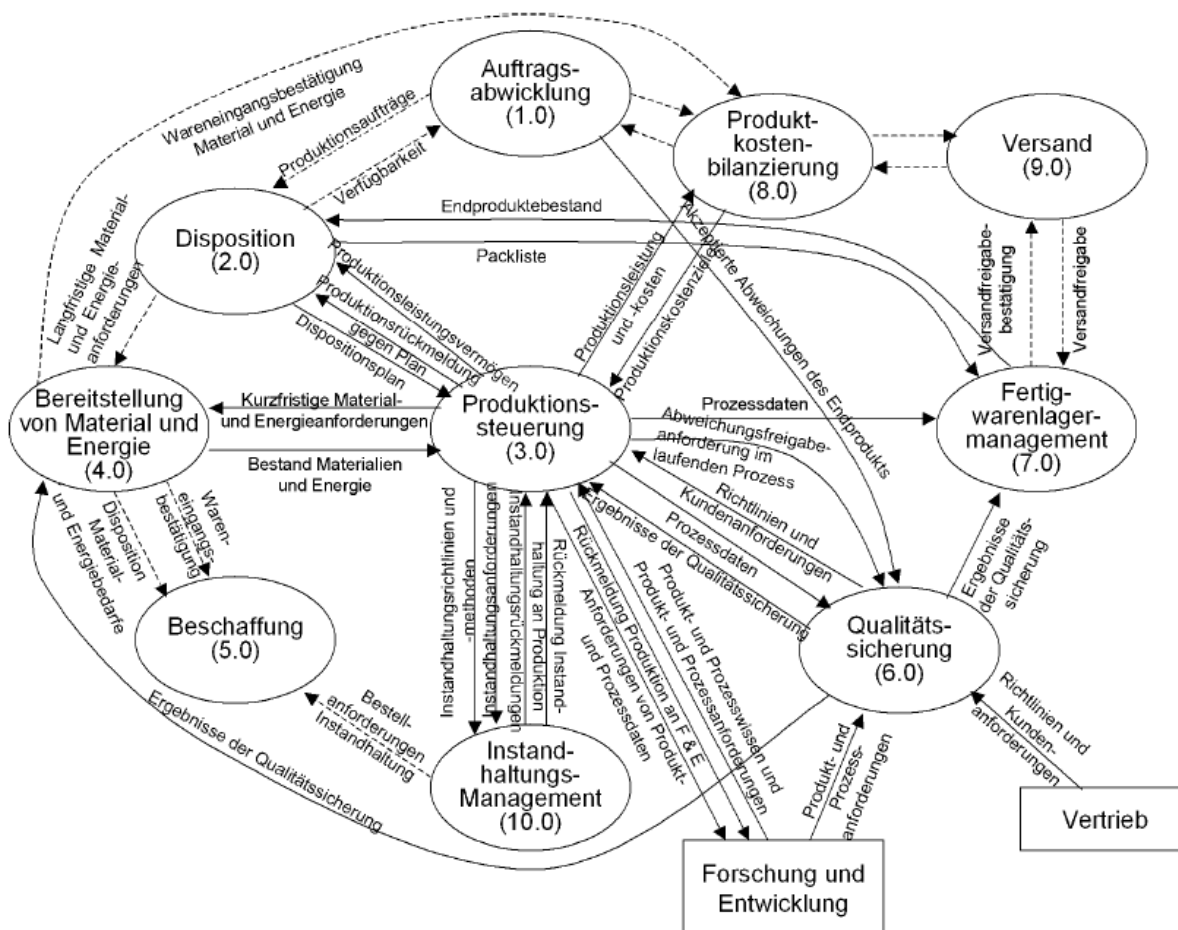


Abbildung 7: Funktionales Unternehmensführungs-/Leitsystemmodell

Jede in Abbildung 7 [15] als Ellipse dargestellte Funktion wird durch ihre Aufgaben näher beschrieben und mittels der Pfeile mit durchgezogener Linie

werden die Datenflüsse zwischen den einzelnen Funktionen dargestellt. Für ERP-Control sind diese Definitionen im Standard besonders interessant, da sie die Tür zu einer prozessorientierten Sichtweise öffnet, welche durch ERP-Control realisiert wird. Die erwähnten Datenflüsse enthalten natürlich auch Daten, welche ebenfalls im Standard beschrieben werden.

Als letzter Abschnitt von Teil 1 des Standards werden unterschiedliche Objektmodelle behandelt. Auch diese sind für ERP-Control von zentraler Bedeutung, da das DB-Schema des Prototyps für die unterschiedlichen Ressourcen des Unternehmens direkt vom Standard abgeleitet wurde und nur in Einzelfällen diesen erweitert. Das Objektmodell gliedert die Arten von Informationen, welche zwischen Managementfunktionen und Betriebsleitung ausgetauscht werden, in drei Kategorien. Das Produktionspotential umfasst alle Informationen, welche die für die Produktion zu Verfügung stehenden Ressourcen beschreiben. Die Produktdefinition beschreibt, wie ein Produkt erzeugt wird. Die Produktionsinformationen beinhalten Daten über die tatsächliche Produktion, also was im Unternehmen hergestellt wird sowie die konkreten Ergebnisse der Produktion.

Die konkreten Ausprägungen der Daten in beispielsweise Materialliste oder Ressourcenliste werden detailliert beschrieben sowie dessen Beziehungen und teilweise auch Überschneidungen untereinander dargelegt. Von besonderem Interesse für ERP-Control sind Definitionen der einzelnen Klassen von Ressourcen, welche mit leichten Modifikationen, um der REA Ontologie von McCarthy zu genügen, im Prototyp umgesetzt wurden. Durch diese Modifikationen wird der Standard allerdings nicht verletzt, da dieser Erweiterungen zulässt um z.B. geschäftsspezifischen Anforderungen Rechnung zu tragen.

Zur näheren Erläuterung sei hier exemplarisch das Materialmodell erwähnt und in Abbildung 8 [15] dargestellt. Dieses Modell wurde mit Ausnahme der Spezifikationen und Ergebnisse der Qualitätssicherung in ERP-Control umgesetzt. Als Material werden im Standard generalisiert alle Formen von Rohstoffen, Zwischen- und Endprodukten angesehen. Eine Materialdefinition beinhaltet hier Eigenschaften eines Materials. Diese Eigenschaften werden in der Materialklasse als konkrete Ausprägung umgesetzt. Das Materiallos spezifiziert die Art und Weise, wie das Material gemessen wird, also gewogen oder gezählt und die dazugehörige Maßeinheit.

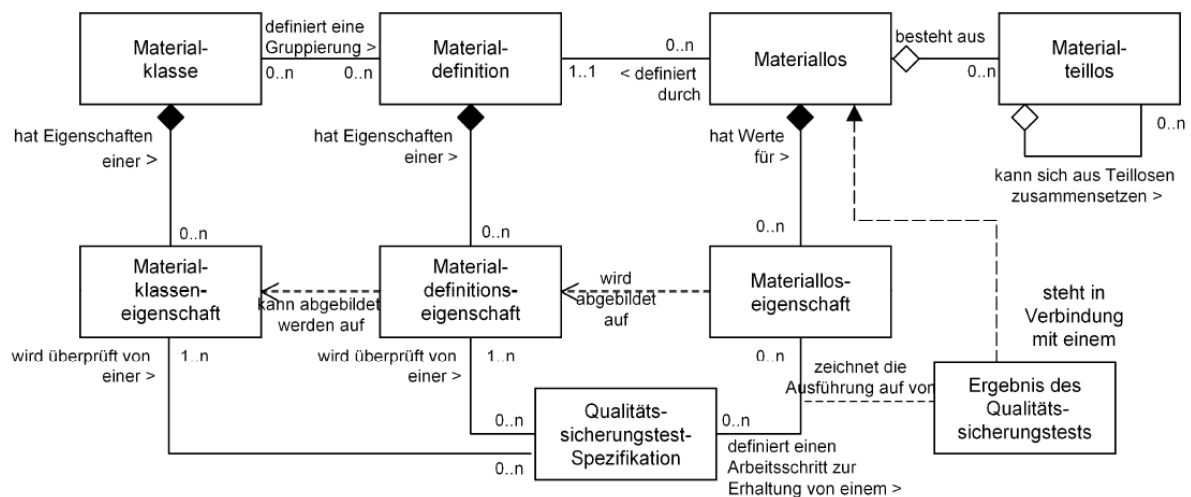


Abbildung 8: Materialmodell

Die konkreten Attribute wie z.B. Maßeinheiten sind das Thema von Teil 2 des ECSI-Standards mit dem Namen „Attribute des Objektmodells“. Die aus Teil 1 bekannten Modelle zu den Unternehmensressourcen werden hier anhand ihrer Attribute näher spezifiziert.

Wie der Titel von Teil 2 „Attribute des Objektmodells“ schon aussagt, liegt hier das besondere Augenmerk auf den Attributen. Diese werden anhand der Objektmodelle aus Teil 1 in Tabellen zusammengefasst, beschrieben und durch Beispiele erläutert. Dadurch soll den Anwendern ein Leitfaden in die Hand gegeben werden, um die im Unternehmen vorhandenen Ressourcen korrekt, vollständig und standardisiert beschreiben zu können. Die Hersteller von Softwarewerkzeugen bekommen dadurch Einblick in die Geschäftsdomäne und können so die Kundenwünsche besser verstehen.

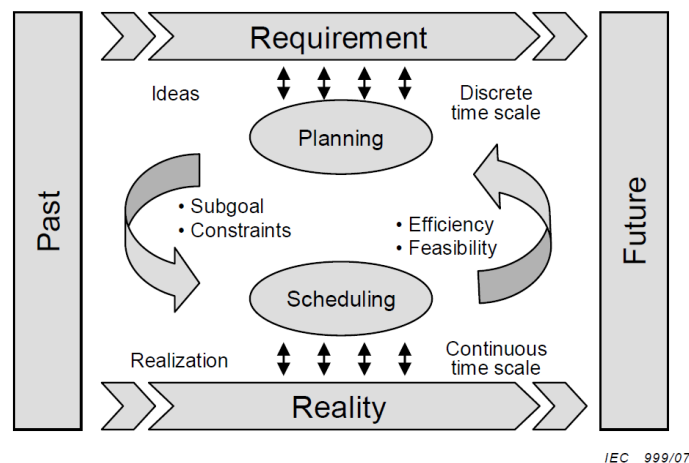
Als Beispiel zur Veranschaulichung soll an dieser Stelle wieder das aus Teil 2 des ECSI-Standards bekannte Materialmodell dienen. Die Attribute der Klasse Materialteillos werden wie in folgender Tabelle 1 [16] beschrieben definiert. An dem Beispiel wird ersichtlich, wie die Attribute einer Klasse zu verwenden sind. Bei der Materialloseigenschaft wird das Material im Attribut ID durch spezifische Eigenschaften wie z.B. Dicke oder Dichte eindeutig identifiziert. Das Attribut Beschreibung definiert die Materialloseigenschaft näher und das Attribut Wert (Value) speichert einen spezifischen Messwert. Im Attribut Messwerteinheit (Value unit of measure) wird schlussendlich die Maßeinheit gespeichert, in der das Attribut Wert gemessen wurde.

Attribute name	Description	Examples
ID	An identification of the specific <i>material lot property</i> .	Average sheet thickness
		Density
Description	Additional information about the <i>material lot property</i> .	Measured thickness
		Measured density
Value	The value, set of values, or range of the property.	5,002
		34,5
Value unit of measure	The unit of measure of the associated property value, if applicable.	mm
		g/cm <sup>3</sup>

**Tabelle 1: Attribute von Materialloseigenschaft**

Auch in diesem Fall bleibt der Standard so generisch wie möglich um geschäftsspezifische Anforderungen ebenfalls zuzulassen.

Teil 3 des ECSI-Standards trägt den Titel „Aktivitätsmodelle für das operative Produktionsmanagement“. Hier werden die in Teil 1 beschriebenen Aktivitäten innerhalb eines Unternehmens durch verschiedene Modelle näher beschrieben. Wie bereits erwähnt nimmt der Standard starken Bezug auf Produktionsunternehmen, weshalb auch in Teil 3 der Schwerpunkt in der Produktion. Das betrifft im vorhin vorgestellten funktionalen Hierarchiemodell Ebene 3 und teilweise darunter, weshalb Teil 3 keinen wesentlichen Bezug zu dieser Diplomarbeit hat. Für diese Arbeit von besonderem Interesse ist allerdings die Hierarchie von Planung und Steuerung wie in Abbildung 9 [17] ersichtlich.



**Abbildung 9: Schematischer Zusammenhang zwischen Planung und Steuerung**

Wie man aus Abbildung 9 ersehen kann, nimmt die Planung (Planning) eher Bezug auf Anforderungen um bestimmte Ziele zu erreichen während die Steuerung (Scheduling) auf die Realität, also erreichte Zustände innerhalb eines

Unternehmens reagiert und gegebenenfalls regelnd eingreift, um geplante Ziele auch zu erreichen. Ein Plan hat also den Zweck, die Umsetzung von Ideen für die Zukunft zu planen und Anforderungen und Ziele für die Produktion zu definieren. Die Steuerung übernimmt dabei eine Kontrollfunktion, indem sie die Anforderungen und Ziele in der Produktion, also der Realität (Reality), überwacht. Die daraus gewonnenen Erkenntnisse können dann wieder für Adaptierungen und Verbesserungen der Planung herangezogen werden, um den Plan besser an die Realität anzupassen.

#### **2.4.3.2. AEO-Standard (ISO/IEC 15944-4)**

Ein weiterer Standard, welcher im Projekt ERP-Control von Bedeutung ist, ist der ISO-Standard 15944-4. Dieser wurde im Jänner 2006 veröffentlicht und trägt den Titel Business Transaction Scenarios – Accounting and economic ontology (AEO). Der Standard greift die REA-Ontologie von McCarthy auf, und versucht auf diese Weise geschäftsrelevante Informationen im Kontext des Datenaustausches aufzubereiten.

Die in diesem Standard verwendete Methode zum Datenaustausch heißt Open-edi und ist ein Standard, welcher im Gegensatz zum weit verbreiteten Edifact-Standard nicht nur Nachrichtentypen und deren Inhalt spezifiziert, sondern häufig auftretende GP und deren Informationsfluss detailliert beschreibt. Als Beispiel seien an dieser Stelle die Phasen einer Geschäftskollaboration (Planung, Identifizierung, Vereinbarung, Aktualisierung, Nach-Aktualisierung) genannt, welche im Standard ISO/IEC 14662 genauer beschrieben werden. Diese Methode folgt der Open-edi Business Transaction Ontology (OeBTO), welche eine formale und regelbasierte Spezifikation und Definition der Transaktionen und Szenarios im Unternehmensbereich sowie die Verbindungen hinter diesen Konzepten beschreibt. Vgl.[9]. Dabei werden die Geschäftspartner in einer solchen Transaktion als unabhängige Ansammlungen von GP betrachtet, zwischen denen ein Austausch von Werten bzw. Ressourcen stattfindet. Letzteres kommt vom Konzept her der REA-Ontologie schon recht nahe. Diese beiden Ontologien werden im AEO-Standard miteinander verknüpft. OeBTO geht in dem Standard dabei eher auf hierarchische Kontrollstrukturen ein, also welche Organisationseinheit Einfluss auf andere ausüben kann und welche Ressourcen die jeweiligen Organisationseinheiten kontrollieren. Dabei werden Organisationseinheiten als auch Ressourcen in einer hierarchischen

Datenstruktur klassifiziert. Diese Art der Klassifizierung findet auch bei ERP-Control statt.

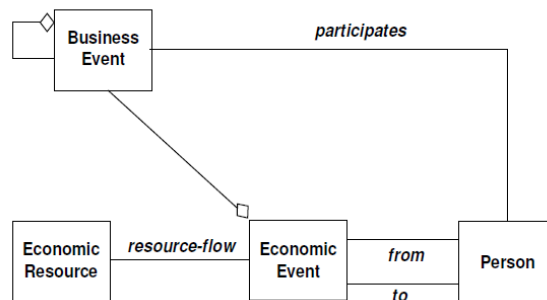


Abbildung 10: REA ergänzt um Business Events

Die REA-Ontologie wird im AEO-Standard durch das in Abbildung 10 [9] gezeigte Konzept der Business Events ergänzt, welche mehrere Economic Events enthalten können. Auch dieses Konzept findet bei ERP-Control Anwendung.

Eine weitere sehr wichtige Spezifikation ist die in Abbildung 11 [9] dargestellte Einführung von Commitments, also Verbindlichkeiten.

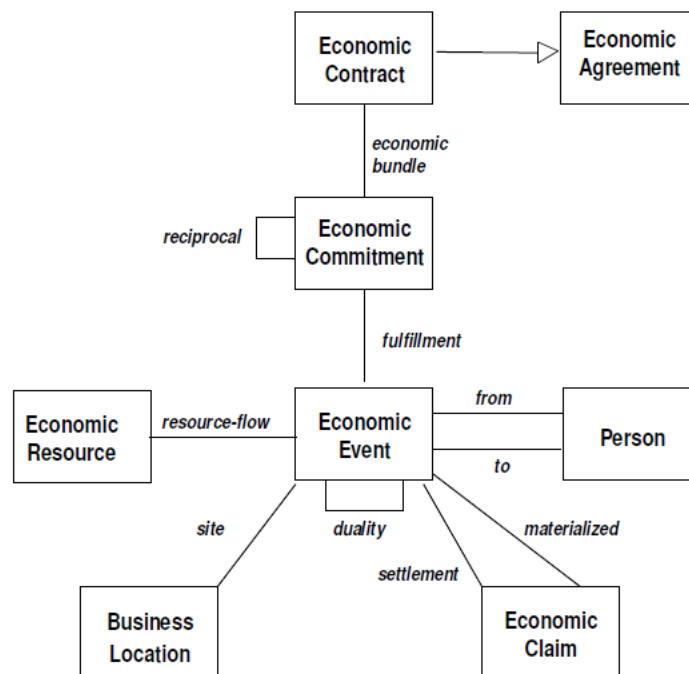


Abbildung 11: Business Transaction Model mit gebündelten Commitments

Diese bilden geplante Ereignisse ab, welche noch nicht realisiert wurden. Die Realisierung erfolgt durch Economic Events. Dementsprechend beinhaltet auch die Spezifikation eines solchen Commitments die Art des Economic Events sowie

die beteiligten Agenten, hier als Rollen deklariert, sowie Ressourcen. Eine Summe von Commitments, sozusagen Leistung und Gegenleistung, kann dabei auch in einem Vertrag (Economic Contract) zusammen gefasst werden. Diese Verträge wiederum können zu Geschäftsvereinbarungen (Economic Agreements) generalisiert werden. Verträge und Vereinbarung werden allerdings nicht vom Standard vorgeschrieben und bleiben daher optional.

### **2.4.3.3. COSO II**

Das Committee of Sponsoring Organizations of the Threadway Commission (COSO) ist eine Vereinigung amerikanischer Privatunternehmen, welche sich zum Ziel gesetzt hat die Methoden zur Unternehmensführung und hier insbesondere der internen Revision zu verbessern. 1992 veröffentlichte dieses Komitee das Rahmenwerk *Internal Control – Integrated Framework*, welches sich mit Controlling und Risikobeurteilung befasste um Unternehmen einen Leitfadens für die Implementierung und Betrieb ihrer internen Kontrollsysteme in die Hand zu geben. Die Erweiterung dieses Rahmenwerks, *Enterprise Risk Managment – Integrated Framework*, ist deren zweite Veröffentlichung weshalb es auch als COSO II bekannt wurde. COSO II hat einen stärkeren Schwerpunkt auf Enterprise Risk Management (ERM), worunter auch die rechtzeitige Erkennung eines risikoreichen Ereignisses gehört.

„Enterprise Risk Management ist ein Prozess, ausgeführt durch die Überwachungs- und Leitungsorgane, Führungskräfte und Mitarbeiter einer Organisation, angewandt bei der Strategiefestlegung und innerhalb der Gesamtorganisation, gestaltet um die Organisation beeinflussende mögliche Ereignisse zu erkennen und um hinreichende Sicherheit bezüglich der Erreichens der Ziele der Organisation zu gewährleisten.“[18]

Die grundlegenden Ziele von COSO II sind folgende:

- Anpassen von Risikoneigung und Strategie [...]
- Verbessern von risikobezogenen Entscheidungen [...]
- Verringern von Überraschungen und Verlusten im Geschäftsbetrieb [...]
- Bestimmen und Steuern mehrfacher und unternehmensübergreifender Risiken [...]
- Nutzen von Chancen [...]
- Verbesserter Kapitaleinsatz „[18]

Nachdem in der SOPPS ein Modell erstellt und kalibriert wird, welche zukünftige positive wie auch negative Entwicklungen in einer Baumstruktur abbildet ist COSO II eine gute theoretische Ergänzung. Wie man anhand der obigen Auflistung erkennen kann, erfüllt die SOPPS die meisten dieser Punkte, da in der stochastischen Planung zukünftige Zustände im Produktionsbereich vorweg genommen werden und damit auch als Chancen oder Risiko erkannt, bewertet und in der Planung berücksichtigt werden können.

Beim Risikomanagement muss dabei bedacht werden, dass es eine direkte Beziehung zwischen den Unternehmenszielen und den Mitteln zur Realisierung ebendieser gibt. Dieser Zusammenhang wird im COSO-Modell als Würfel wie in Abbildung 12 [18] dargestellt.

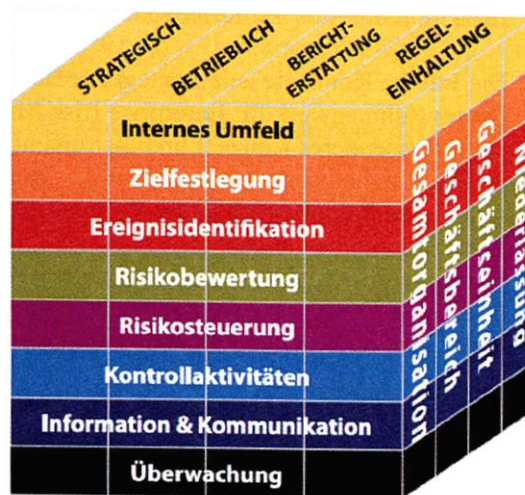


Abbildung 12: COSO-Würfel

Diese Ziele bezüglich ERM werden in COSO II in folgende Kategorien unterteilt, um sie den verschiedenen Bedürfnissen der unterschiedlichen



Organisationseinheiten besser zuordnen zu können und damit auch Verantwortlichkeiten abzuleiten:

- Strategische Ziele
- Operative Ziele
- Zuverlässige Berichterstattung
- Regeleinhaltung (Compliance) Vgl.[18]

Neben strategischer und operativer Ziele sind bei ERM besonders eine akkurate Berichterstattung zur Risikoerfassung sowie die Einhaltung geltender Gesetze und Vorschriften von besonderer Bedeutung. Die Mittel zur Realisierung dieser Ziele wird nun im Risikomanagement abgebildet und durch folgende Komponenten beschrieben, welche im Managementablauf integriert sein sollten:

- Internes Umfeld (Unternehmenskultur, Risikophilosophie)
- Zielfestlegung
- Ereignisidentifikation (Chancen und Risiken)
- Risikobewertung (Bewertungsgrundlage für Risikosteuerung)
- Risikosteuerung
- Kontrollaktivitäten (Vorschriften und Verfahren für wirksames ERM)
- Information und Kommunikation
- Überwachung

Das interne Umfeld ist hierbei von besonderer Bedeutung. Es beschreibt die Einstellung der Mitarbeiter eines Unternehmens zum Thema Risiko. Nur mit einer entsprechenden Unternehmenskultur und einer daraus gewachsenen Philosophie im Umgang mit Risiken können Mitarbeiter auf dieses Thema sensibilisiert werden. Das beeinflusst auch die Zielfestlegung, also die Definition der strategischen Ziele eines Unternehmens. Wenn das Management nicht auf Risiko sensibilisiert ist, werden diesem unter Umständen nicht jene Bedeutung zugemessen, die es „verdient“.

Bei der Ereignisidentifikation geht es darum zu erkennen, dass ein Ereignis sowohl Chancen als auch Risiken in sich bergen kann. Diese Ereignisse müssen natürlich auch in der Hinsicht bewertet werden, welchen Einfluss sie auf die vorab definierten Ziele haben. Der nächste Schritt ist die Steuerung von Risiken, also wie diese im Falle eines Eintretens bewältigt werden können. Ein Restrisiko lässt sich dabei in den seltensten Fällen ausschließen.

Die Risikobewältigung führt direkt zum Thema Kontrollaktivitäten. Es muss mit Vorschriften und Verfahren sichergestellt werden, dass die Risikosteuerung auch funktioniert. Im Informationszeitalter kaum mehr weg zu denken ist der Punkt Information und Kommunikation. Für eine effiziente Zielerreichung ist der Informationsfluss zwischen den Organisationshierarchien essenziell. Daran angeschlossen ist auch das Berichtswesen, welches dem Unternehmen hilft Fehler zu verstehen oder innovative Änderungen eines GP zu dokumentieren und zu replizieren. Ohne eine intakte Kommunikation und Informationsverbreitung sinkt auch das Risikobewusstsein der Mitarbeiter. Hier trifft das Sprichwort „Aus den Augen – aus dem Sinn!“ dann tatsächlich zu. Schlussendlich impliziert ein Berichtswesen auch den Punkt Überwachung, ohne den es ja nichts zu berichten gäbe.

Die dritte Dimension des Würfels identifiziert die Organisationseinheiten und will so die Möglichkeit visualisieren, ERM unternehmensweit oder tiefer in der Hierarchie umzusetzen bzw. die Tatsache zu unterstreichen, dass die unterschiedlichen Komponenten des Risikomanagements in unterschiedlichen Hierarchiestufen umgesetzt werden.

Des Weiteren werden in COSO II die einzelnen Rollen und Verantwortlichkeiten der Führungsorgane sowie der Mitarbeiter in Bezug auf ERM beschrieben. Diese Beschreibungen beinhalten nicht nur innerbetriebliche Organisationseinheiten, sondern gehen auch Gesetzgeber, Aufsichtsbehörden und Prüfungsorganen oder Geschäftspartner wie Kunden, Lieferanten und Outsourcing-Partnern ein.

## 2.4.4. Relevante Konzepte

Nachfolgend werden einige Konzepte, welche im Umfeld von ERP-Systemen und speziell im Bereich der Planung und des Controllings eine nicht unbedeutende Rolle spielen. Dieser Abschnitt dient einer Begriffsdefinition, da manche Konzepte in diesem Bereich durchaus missverstanden werden können oder nicht eindeutig definiert sind. Des Weiteren wird auch das Hierarchiemodell von Mihajlo D. Mesarovič vorgestellt, der im Bereich der Systemtheorie wegweisende Arbeit geleistet hat.

### 2.4.4.1. Strategische Planung

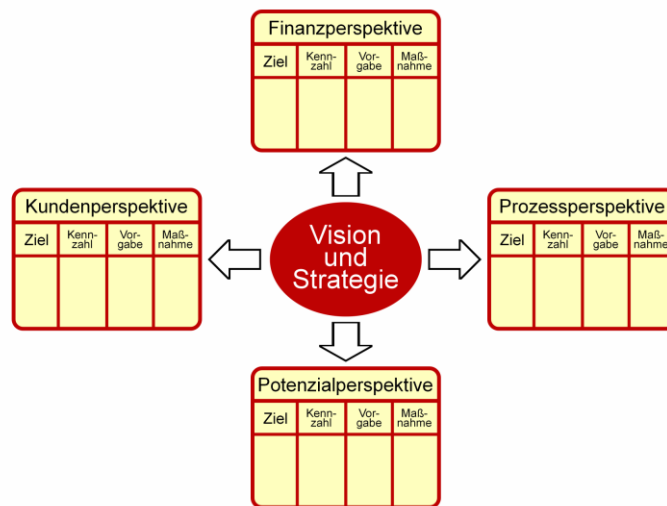
In der Literatur finden sich zu diesem Begriff unterschiedliche Meinungen und Definitionen. Ursächlich dafür ist die etwas verwirrende Wortkombination, da eine Strategie und ein Plan im Wesentlichen dieselben Konzepte sind. Der Unterschied besteht 1) im Zeitraum, der geplant wird und 2) dem Ziel des Plans. Eine Strategie ist langfristig ausgelegt und muss daher mehrere Szenarien berücksichtigen, von denen jeweils eines als Optimum aus dem Planungsprozess hervorgeht, oder ein Szenario im Endeffekt realisiert wird. Man bereitet sich also auf mehrere Möglichkeiten vor. Eng damit verknüpft ist die Vision eines Unternehmens, also Ziele auf die das Unternehmen längerfristig hin arbeitet. „Eine Strategie dient mit anderen Worten der Realisierung einer Vision.“ [19] Im Falle der strategischen Produktionsplanung besteht die Vision darin, wie viel ein Unternehmen im Laufe des Planungszeitraumes produzieren und absetzen soll.

Eine Strategie wird auch gerne als grober Leitfaden verstanden. Das ist nicht gänzlich falsch, allerdings geht es dabei nicht nur um eine grobe Marschrichtung, sondern darum, Ziele zu verfolgen und diese mitunter auf unterschiedlichen Wegen, die nicht im Vorhinein festgelegt sind. „Mit anderen Worten: *Der Weg ist das Ziel!*“ [19] Im Gegensatz dazu wird ein Plan eher im operativen Bereich gesehen. Hier weiß man schon ganz konkret, was zu tun ist und plant auf dieses Ziel konkret hin. Man kennt den Weg also schon.

Die strategische Planung harmoniert daher sehr gut mit stochastischen und kybernetischen Konzepten. Aus im SOPPS dieser Arbeit wird nicht ein konkreter Weg zum Ziel hin definiert, sondern vielmehr ein Handlungsspielraum definiert, der mehrere Wege offen lässt, ohne das Ziel aus den Augen zu verlieren.

## 2.4.4.2. Balanced Score Card

Im Kontext von Strategie und Vision soll auch die Balanced Score Card (BSC) an dieser Stelle kurz Erwähnung finden. Die BSC stellt ein Rahmenwerk da, um die Unternehmensstrategie anhand von Kennzahlen messbar zu machen. Wie in der Abbildung 13 ersichtlich werden dazu meist vier unterschiedlichen Perspektiven unterschieden, obwohl es mittlerweile auch für diese Einteilung modifizierte Versionen und Derivate existieren.



Quelle: <http://de.wikipedia.org>

Abbildung 13: Balanced Score Card

In der Finanzperspektive wird z.B. der Gewinn erfasst. Die Kundenperspektive stellt z.B. durch die Kennzahl Kundenzufriedenheit einen Kontext zur Finanzperspektive her. Die Prozessperspektive kann sich unter anderem auch mit der Qualität der Produktion in Form einer Kennzahl für die Ausschussrate befassen, was wiederum direkten Einfluss auf die Kundenzufriedenheit hat. Um die Qualität zu steigern bedarf es gut ausgebildeter Mitarbeiter. Ihr Wissensschatz trägt dazu bei, die Qualität in der Produktion zu gewährleisten. Vgl. [20].

### 2.4.4.3. Hierarchiemodell nach Mesarovič

Der serbische Elektroingenieur Mihajlo D. Mesarovič gilt als einer der Pioniere im Forschungsfeld der Systemtheorie. Im Laufe seiner akademischen Laufbahn war er in diversen Forschungsstätten aktiv, darunter das Nikola-Tesla-Institut in Belgrad, die Universität Belgrad oder am MIT<sup>2</sup>. In seinem Buch *Theory of Hierarchical, Multilevel, Systems* [21] beschreibt er hierarchische Systeme unter den Gesichtspunkten Abstraktion in der Modellbildung, Komplexität in der Entscheidungsfindung sowie Prioritätshierarchien in Entscheidungssystemen mit mehreren untereinander konkurrierenden Elementen, in konzeptioneller sowie mathematischer Weise. Ein Hauptaugenmerk seiner Ausführungen liegt dabei in der Koordination der einzelnen Teile eines Systems. Die Prioritätshierarchien sind für diese Arbeit dabei weniger relevant, da es zu keinen Kompetenz- und Zielkonflikten unter den einzelnen Elementen kommt.

In seinem Buch beschäftigt sich Mesarovič auch mit der Lösung komplexer Probleme unter Zuhilfenahme eines rekursiven Ansatzes, der ein komplexes Problem in viele kleine und einfacher lösbare Probleme aufteilt, die in einer hierarchischen Anordnung in Summe die Lösung des Ursprungsproblems ergeben. Er beschreibt dabei Entscheidungs- oder Steuerungssysteme und eine stochastisch optimale Planung und Steuerung ist dabei seinen Modellen nicht unähnlich. Ein ähnliches Prinzip findet sich auch beim kybernetischen Steuerungsprinzip wieder.

Mesarovič erläutert dabei, dass eine Hierarchie in solchen Systemen auf Grund der Problemstellung unter Unsicherheit auf natürlichem Wege zustande kommt. Dabei nennt er drei wesentliche Aspekte:

- ”
- i) the selection of strategies to be used in the solution process  
[...]
  - ii) the reduction or elimination of uncertainties [...]
  - iii) the search for a preferable or acceptable course of action  
under prespecified conditions [...]“ [21]

Punkt i) beschreibt dabei die Berechnung einer Entscheidung basierend auf einer Auswahl möglicher Handlungsweisen auf dem System übergebenen Informationen und einer Evaluationsfunktion, welche eine Handlungsweise aus

---

<sup>2</sup> Massachusetts Institute of Technology

der Auswahl bestimmt. Punkt ii) stellt einen Lernprozess, welcher die Auswahl an möglichen Handlungsweisen weiter eingrenzen soll. Punkt iii) definiert Strukturen, Funktionen und Strategien, welche in den vorhergehenden Punkten angewandt werden.

Durch diesen Ansatz wird bei der Modellbildung, wo die Ausgangssituation sowie die Problemstellung durch die Zerlegung in Teilprobleme abstrahiert und die Komplexität des Problems herabgesetzt wird. Ein ähnlicher Ansatz kommt auch im praktischen Teil dieser Diplomarbeit zum Einsatz. Nachdem eine stochastisch optimale Planung und Steuerung keine triviale Aufgabe ist, wird diese in kleinere Teilprobleme zerlegt. Analog zu Mesarovič Beispiel der Entscheidungshierarchie wird

- i) ein Wahrscheinlichkeitsraum von möglichen Zuständen aufgespannt, welcher durch eine Evaluationsfunktion eine optimale Handlungsweise, hier Produktionsmenge, bestimmt,
- ii) durch die optionale Verwendung historischer Daten wird die Konstruktion des Wahrscheinlichkeitsraumes optional auf Erwartungswerte für die Schwankungsbreite der Nachfrage und Anteil der einzelnen Perioden am Gesamtumsatz hin kalibriert sowie unwahrscheinliche Zustände im Laufe der Steuerung eliminiert und
- iii) die Struktur des Wahrscheinlichkeitsraumes, hier der Entscheidungsbaum, wird definiert sowie Funktionen bereit gestellt, um die einzelnen Elemente des Wahrscheinlichkeitsraumes mit Werten zu befüllen und mit diesen auch zu agieren.

Alle drei Elemente des Systems sind voneinander abhängig und tauschen Informationen aus, die das jeweils andere Element benötigt um seine Aufgabe zu erfüllen. Jedes Element für sich alleine kann die Problemstellung nicht bewältigen, aber durch die hierarchische Herangehensweise und die dadurch bewirkte Zerlegung der Problemstellung in Teilprobleme ist selbige auch lösbar.

## 2.5. Mathematische Grundlagen

Eines der zentralen Aufgaben in der Wirtschaftsinformatik ist es ja, ökonomische Problemstellungen durch Methoden und Anwendungen der IT zu lösen. Dabei kommen weder Ökonomen noch Informatiker ohne Mathematik aus, zumal die Anfänge des Computers namensgebend als Rechenmaschinen dienten. Nachfolgend werden jene Konzepte erläutert, welche zur Realisierung einer SOPPS angewandt werden.

### 2.5.1. Umsatzergebnisfunktion

Um den Erfolg einer Unternehmung, in diesem Falle der Produktion, bewerten zu können, muss man im Wesentlichen zwei Dinge wissen. Was kostet die Produktion und welchen Erlös kann man daraus erwirtschaften. Aus diesen Überlegungen heraus kann man sich ein mathematisches Modell formen, welches eben diese beiden Parameter berücksichtigt. Die nachfolgenden Ausführungen sind stark an die Vorlesung *Controlling unter Unsicherheit* von Prof. Schwaiger und dem dazugehörigen Skriptum [22] angelehnt.

Die Kosten teilen sich dabei in variable und fixe Kosten auf. Variable Produktionskosten ( $k_{v,p}$ ) berücksichtigen dabei jene Kosten, welche direkt der Herstellung einer Einheit des Produktes zugeordnet werden können. Das können z.B. Produktionseinsatzfaktoren wie Personalkosten oder Materialkosten sein. Im Falle einer Überproduktion bleibt ein Anteil der Produktion als Lagermenge ( $X_{L,t}$ ) zurück und es fallen auch Lagerkosten ( $k_{v,L,t}$ ) an, welche ebenfalls zu den variablen Kosten gezählt werden. Die Lagermenge ist dabei wie folgt definiert:

$$\tilde{X}_{L,t} = \max(X_{P,t} + X_{L,t-1} - \tilde{X}_{N,t}; 0)$$

Formel 1: Definition Lagermenge

Sie ergibt sich aus der produzierten Menge ( $X_{P,t}$ ) des Betrachtungszeitraums zusammen mit einer eventuellen Lagermenge aus der Vorperiode ( $X_{L,t-1}$ ) abzüglich der Nachfragemenge. Unter Umständen kann das Lager aber auch leer sein. Nachdem die Nachfragemenge ( $\tilde{X}_{N,t}$ ) zum Zeitpunkt der Produktion nicht bekannt ist und die Lagermenge von ebendieser abhängt, werden beide

Variablen als Zufallsvariablen angesehen. Dies wird durch die Tilde, welche sich als Akzent über der Variablenbezeichnung befindet, gekennzeichnet.

Im Gegensatz dazu fallen noch Fixkosten ( $K_F$ ) an, welche unabhängig von der Produktionsmenge anfallen und daher als statisch angesehen werden können. Darunter versteht man z.B. Kosten für Anlagen, Immobilien oder Verwaltung, welche in jedem Fall entstehen auch wenn nichts produziert wird.

Diesen beschriebenen Ausgaben stehen Einnahmen gegenüber, welche ebenfalls variabel definiert sind und sich aus einem Verkaufspreis ( $p$ ) multipliziert mit der erzielten Absatzmenge ( $\tilde{X}_{Abs,t}$ ) zusammensetzen. Die Absatzmenge wird dabei folgendermaßen bestimmt:

$$\tilde{X}_{Abs,t} = \min(\tilde{X}_{N,t}; X_{P,t} + X_{L,t-1})$$

**Formel 2: Definition Absatzmenge**

Die Absatzmenge muss nicht gezwungenermaßen der Nachfragemenge entsprechen. Wird die Nachfrage durch die Produktion nicht befriedigt, so setzt sich diese aus der Produktionsmenge des Betrachtungszeitraumes sowie der Lagermenge aus der Vorperiode zusammen. Im umgekehrten Fall einer Überproduktion entspricht die Absatzmenge der Nachfragemenge, da selbstredend nicht mehr abgesetzt werden kann, als nachgefragt wird.

Diese Gegenüberstellung lässt sich in der Umsatzergebnisfunktion beschreiben, wobei Produktionsmenge und Absatzmenge unter Einbeziehung stochastischer Überlegungen als Zufallsvariablen definiert sind.

Die Gesamtkosten ( $\tilde{K}$ ) lassen sich also anhand folgender Formel bestimmen:

$$\tilde{K} = k_{v,P} * X_{Abs,t} + k_{v,L,t} * \tilde{X}_{L,t} + K_F$$

**Formel 3: Gesamtkosten**

Die Einnahmen lassen sich als Umsatzfunktion wie folgt definieren:

$$\tilde{UMS} = \tilde{X}_{Abs,t} * p$$

**Formel 4: Umsatz**



Beides zusammen ergibt die Umsatzergebnisfunktion, wobei man den Verkaufspreis abzüglich der variablen Produktionskosten als Deckungsbeitrag ( $db_t$ ) definieren kann, da ja nur verkaufte Produkte zur Wertschöpfung beitragen.

$$\begin{aligned}\tilde{U}E &= \tilde{U}MS - \tilde{K} \\ \tilde{U}E &= (p - k_{v,p}) * \tilde{X}_{Abs,t} - k_{v,L,t} * \tilde{X}_{L,t} - K_F \\ \tilde{U}E &= db_t * \tilde{X}_{Abs,t} - k_{v,L,t} * \tilde{X}_{L,t} - K_F\end{aligned}$$

**Formel 5: Herleitung Umsatzergebnis**

Wie bereits erwähnt dient diese Formel zur Erfolgsmessung einer Unternehmung. Genau zu diesem Zweck wird sie auch in der stochastisch optimalen Planung und Steuerung eingesetzt. Durch die Einbezugnahme von sowohl internen als auch externen Faktoren kann aus unternehmerischer Sicht ein Zustand innerhalb einer intertemporal optimierten Planung durch eine Kennzahl ermittelt werden. Die Produktionsmenge stellt dabei die Stellgröße dar, mit Hilfe derer in der Produktionssteuerung der Umsatzerfolg optimiert werden soll. Durch einen enumerativen Algorithmus werden alle möglichen Produktionsmengen simuliert und das maximale und damit optimale Ergebnis ermittelt.

### **2.5.2. Stochastische Prozesse und Filtrierte Wahrscheinlichkeitsräume**

Der Kampf gegen den Zufall, wie es Albert Ducrocq bei der kybernetischen Steuerung nannte fügt sich fast nahtlos in das Fachgebiet der Stochastik ein, welche als Teilgebiet der Mathematik die Wahrscheinlichkeitstheorie und Statistik beinhaltet. Auch die SOPPS bedient sich dieses Fachgebietes, um die Zukunft nicht dem Zufall zu überlassen, sondern in gewissen Grenzen vorhersagbar zu machen. Der folgende Abschnitt soll kurz in das Themengebiet einführen und insbesondere auf Wahrscheinlichkeitsräume und deren Filtration eingehen. Dieses Konzept dient dazu die zukünftigen Zustände eines Systems, in diesem Fall der Marktsituation und wie die Produktion darauf reagiert, auf ein mathematisch solides Fundament zu stellen und abzubilden.

### Wahrscheinlichkeitsraum:

Zur Darstellung von Wahrscheinlichkeitsräumen wird die  $\sigma$ -Algebra verwendet. Eine  $\sigma$ -Algebra bezeichnet in der Stochastik ein System ( $A$ ) von Teilmengen ( $\Omega = \{\omega_1, \omega_2 \dots \omega_i\}$ ), wenn folgende Bedingungen erfüllt sind:

- ”
- $\Omega \in A$ ,
  - $E \in A \Rightarrow E^c \in A$ ,
  - $E_1, E_2, E_3 \dots \in A \Rightarrow \bigcup_{i=1}^{\infty} E_i \in A$ .“ [1]

#### Formel 6: Definition $\sigma$ -Algebra

$\Omega$  repräsentiert hierbei den Stichprobenraum oder im Falle der Produktionsplanung jene Pfade, die im postulierten Modell beschriftet werden können. Für die spätere Definition des stochastischen Prozesses wird auch die Potenzmenge  $P(\Omega)$  wichtig. Diese Menge stellt die Summe aller Teilmengen von  $\Omega$  dar. Die einzelnen Elemente von  $\Omega$  werden mit einem Index unterschieden und als  $\omega$  bezeichnet.  $A$  beinhaltet all jene „Teilmengen von  $\Omega$ , für die wir eine Wahrscheinlichkeit messen können wollen.“ [1] Die Mengen  $E \in A$  stellen die Ereignisse bzw. messbaren Mengen dar.

### Stochastischer Prozess:

Um auf dieser  $\sigma$ -Algebra im Zeitverlauf operieren zu können ist ein stochastischer Prozess von Nöten. Dieser wird von Gerhard Hübner folgendermaßen definiert:

„Für einen stochastischen Prozess benötigt man „ einen Wahrscheinlichkeitsraum „ $(\Omega, A, P)$ , einen Bildbereich  $\Omega'$ , der auch Zustandsraum oder Zustandsmenge heißt, einen Zeitbereich  $T$  [...] und für den Zeitpunkt  $t \in T$  eine Zufallsvariable  $X_t: \Omega \rightarrow \Omega'$ . Dann heißt  $(X_t) := (X_t, t \in T)$  ein stochastischer Prozess.“[23]

### Zufallsvariable:

Hübner meint mit einer Zufallsvariable allerdings nicht eine zufällig gewählte Größe. Diese Namensgebung beruht auf den mittlerweile veralteten Gebrauch dieses Begriffs für „abhängige Variable“ und meint eigentlich eine Abbildung, also eines Ereignisses innerhalb von  $\Omega$ , dass durch eine Zufallsvariable beschrieben wird. Mit anderen Worten ist eine Zufallsvariable in diesem Kontext eine Größe,

die nicht vorab bestimmt werden kann und erst im Zeitverlauf durch Informationsenthüllung determiniert wird.

Ein stochastischer Prozess ist nicht deterministisch, daher wohnt ihm ein gewisses Zufallsmoment inne. Erst nach der Realisierung einer Zufallsvariable erhält diese auch einen konkreten Wert. „Ein stochastischer Prozess ist demnach als eine Abbildung zu verstehen, die jedem Ergebnis  $\omega$  eines Zufallsvorganges [...] eine Zeitreihe zuordnet.“ [24]

Dazu definiert Heinz Bauer  $\omega$  als Abbildung folgendermaßen:

„Für jedes  $\omega \in \Omega$  heißt die durch  $t \rightarrow X_t(\omega)$  definierte Abbildung „vom Zeitbereich im Zustandsraum „ein Pfad des Prozesses. [...] Statt Pfaden spricht man auch von Trajektorien oder Realisierungen des Prozesses.“ [25]

Das heißt, dass der stochastische Prozess nicht als Aneinanderreihung von Zufallsvariablen gesehen werden soll, sondern als eine Menge von Realisierungen innerhalb von  $\Omega$ . Erst durch diese Betrachtungsweise wird der prozessorientierte Charakter des stochastischen Prozesses deutlich. Der Ereignisraum enthält eine Menge von Trajektorien, die erst im Zeitablauf realisiert werden. Diese Einsicht führt direkt zur Datenstruktur eines Baumes, welche im folgenden Kapitel erläutert wird.

### **Random Walk:**

Einen Sonderfall von stochastischen Prozessen stellen Random Walks (RW) dar, welche an dieser Stelle nicht unerwähnt bleiben dürfen. Der Name rührt daher, dass es sich bei der Realisierung einer Trajektorie um eine unvorhersehbare Zufallsbewegung handelt. Die Zufallsvariablen eines RW verändern sich dabei im Zeitverlauf anhand einer unabhängigen und identischen Verteilung (iid-Bedingung) was auch die Definition eines RW ist. Diese RWs lassen sich dabei in geometrische und additive Prozesse unterscheiden wobei die Art der Veränderung unterschieden wird, anhand derer sich die Zufallsvariablen im Zeitverlauf entwickeln. Der Unterschied besteht in der Wertbelegung der einzelnen Zustände entlang der Trajektorien. Betrachtet man den Wahrscheinlichkeitsraum nicht als eine Ansammlung wahrscheinlicher

Ergebnismengen von  $\omega$  entlang der Trajektorien sondern als einzelne Zustände, ergibt sich ein Zustandsraum wie in Tabelle 2 zu sehen.

$\Omega \times T$	$t_0$	$t_1$	$t_2$	$t_3$
$\omega_1$	$s_{0,1}$	$s_{1,0}$	$s_{2,0}$	$s_{3,0}$
$\omega_2$				$s_{3,1}$
$\omega_3$			$s_{2,1}$	$s_{3,2}$
$\omega_4$				$s_{3,3}$
$\omega_5$		$s_{1,1}$	$s_{2,2}$	$s_{3,4}$
$\omega_6$				$s_{3,5}$
$\omega_7$			$s_{2,3}$	$s_{3,6}$
$\omega_8$				$s_{3,7}$

**Tabelle 2: Zustandsraum**

Bei geometrischen Random Walks kommt dabei eine Multiplikation zum Einsatz, um die Werte der einzelnen Zustände  $s_{t,i}$  zu modellieren. Nach der Berechnung

$$s_{t,i} = s_{t-1,i} * \tilde{z}_j$$

**Formel 7: Zustand im geometrischen RW**

ergibt sich der Wert eines  $s_{t,i}$  stets aus dessen Vorgänger multipliziert mit einem bestimmten Wert  $\tilde{z}_j$ , der meistens einen Prozentsatz repräsentiert und dem Zustand  $s_{t,i}$  einen höheren oder niedrigeren Wert im Vergleich zu  $s_{t-1,i}$  gibt. Dieses Modell ist besonders in der Finanzmathematik sehr verbreitet. Ein bekanntes Beispiel ist das Black-Scholes-Modell zur Bewertung von Aktienoptionen. Bei der Bewertung spielt der risikofreie Zinssatz eine wesentliche Rolle. Der Wert der Option berechnet sich dabei pro Periode nach obiger Berechnung, wobei der Zinssatz das Zufallselement darstellt.

Ein anderer Vertreter aus der Familie der RWs ist der additive Random Walk. Dabei entwickeln sich  $\omega$  im Zeitverlauf nicht geometrisch sondern eben additiv ermittelt. Die Werte der Zustände  $s_{t,i}$  werden ausgehend von einem Ausgangswert  $a_t$  und einem Wert  $\tilde{x}_j$  nach Berechnung

$$s_{t,i} = a_t + \tilde{x}_j$$

**Formel 8: Zustand im additiven RW**

angenommen. Der Ansatz in diesem Fall ist ein völlig anderer. Anstatt dass sich  $s_{t,i}$  im Zeitverlauf anhand von Multiplikationen weiter entwickelt, wird im hier verwendeten Modell ausgehend dem Ausgangswert  $a_t$  der Wert  $\tilde{z}_j$  addiert.

Die Unterschiede zwischen additiven und geometrischen RW sollten nun deutlich erkennbar sein. Im geometrischen RW werden die Zufallsvariablen anhand der vorhergehenden Zufallsvariable im Zeitablauf bestimmt. Das hat zur Folge, dass die Zufallsvariablen mit zunehmenden  $t$  immer stärker streuen während im additiven RW die Veränderung von  $s_t$  zu  $s_{t+1}$  durch eine Addition anstatt einer Multiplikation bestimmt wird und somit immer innerhalb der Wertbelegung von  $\tilde{z}_j$  streuen. Der additive RW hat zudem den Vorteil, dass jede Periode in  $T$  separat und unabhängig von der Vorperiode modellieren lässt. Dadurch können auch saisonale Effekte bei der Nachfrageentwicklung abgebildet werden, weshalb dieses Modell auch in dieser Arbeit zur Anwendung kommt. Dabei muss allerdings beachtet werden, dass eine separate Modellierung der einzelnen Perioden die iid-Bedingung verletzt, womit es sich bei einem solchen Modell nicht mehr um einen RW handelt. Der Prototyp dieser Arbeit ermöglicht eine solche Modellierung, konzeptionell rühren die zugrundeliegenden Überlegungen allerdings von einem RW-Ansatz her.

### **Filtrierung des Wahrscheinlichkeitsraumes:**

Die Filtrierung dieses nun definierten Wahrscheinlichkeitsraumes findet durch eine zeitdiskrete Partitionierung statt. Jedem Zeitpunkt innerhalb des Wahrscheinlichkeitsraumes ist eine Teilmenge von  $\Omega$  zugeordnet. Anstatt eine Folge von einzelnen Zufallsvariablen zu verwenden wird hier eine oder mehrere Mengen einem Zeitpunkt zugeordnet. Im Zeitpunkt 0 beträgt diese Menge  $\Omega$ . Beispielsweise würde sich diese Menge unter der Annahme, dass der Partitionierung ein binärer Baum zugrunde liegt, in der nächsten Periode in zwei gleich große Mengen aufspalten. In der letzten Periode, also wenn vollkommene Informationsenthüllung vorliegt, ergibt sich die Anzahl der einzelnen Teilmengen aus der Anzahl der Elemente von  $\Omega$ . Jedem Zustand kann somit ein konkreter Wert zugeordnet werden.

Zur Illustration stelle man sich eine Baumstruktur wie in folgender Tabelle 3 [22] vor.

$\Omega \times T$	$t_0$	$t_1$	$t_2$	$t_3$
$\omega_1$	$\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6, \omega_7, \omega_8\}$	$\{\omega_1, \omega_2, \omega_3, \omega_4\}$	$\{\omega_1, \omega_2\}$	$\{\omega_1\}$
$\omega_2$				$\{\omega_2\}$
$\omega_3$			$\{\omega_3, \omega_4\}$	$\{\omega_3\}$
$\omega_4$				$\{\omega_4\}$
$\omega_5$		$\{\omega_5, \omega_6, \omega_7, \omega_8\}$	$\{\omega_5, \omega_6\}$	$\{\omega_5\}$
$\omega_6$				$\{\omega_6\}$
$\omega_7$			$\{\omega_7, \omega_8\}$	$\{\omega_7\}$
$\omega_8$				$\{\omega_8\}$

Tabelle 3: Partitionierung des Wahrscheinlichkeitsraumes

Die oben dargestellte  $\sigma$ -Algebra verfügt über einen Wahrscheinlichkeitsraum von acht Elementen  $\omega_1 - \omega_8$ . In jeder Periode wird ein Zustand realisiert wodurch der Ergebnisraum um eben jene Elemente beschnitten wird, die nicht mehr wahrscheinlich sind. In der letzten Periode herrscht vollkommene Informationsenthüllung und man weiß genau welches Element des Wahrscheinlichkeitsraumes realisiert wurde.

### 2.5.3. Rekursiver Lösungsansatz für das SOPPS

Im Gegensatz zu einem deterministischen System, wo sämtliche Zustände bereits im Vorfeld bekannt sind, werden bei stochastischen Systemen Informationen und damit auch Zustände erst im Zeitverlauf enthüllt. Das hat zur Konsequenz, dass vorab festgelegte Stellgrößen wie in einem deterministischen System in einem stochastischen System unter Umständen nicht den gewünschten Lenkungseffekt bewirken, da hier auch unerwartete Zustände auftreten können und mit starren Stellgrößen darauf nicht reagiert werden kann. Vgl. [26]. Dieses Problem erinnert an den Unterschied zwischen Plänen und Strategien oder kybernetischer Regelung und Steuerung.

Genau dieses Thema adressiert die Stochastische Kontrolltheorie. Anstatt retrospektiv auf einen Messwert mit der Regelung einer Stellgröße zu reagieren, muss in der stochastischen Kontrolltheorie flexibel auf die Zustände des Systems

reagiert werden können. Einer Reaktion geht eine Aktion voraus, deshalb ist das im eigentlichen Sinne der falsche Terminus. Wie im vorhergehenden Abschnitt dargelegt wurde, geht es bei stochastischen Prozessen ja um die Filtrierung des Wahrscheinlichkeitsraumes und damit auch darum, zukünftige Ereignisse abschätzen zu können.

Dies führt zu der Überlegung, einen Optimierungsansatz für die Auswahl einer Stellgröße zu wählen, der mehrere Perioden berücksichtigt. Markos Papageorgiou (siehe [27]) sieht einen solchen Optimierungsansatz in der Verwendung von Erwartungswerten. Nachdem sich der vorhin erläuterte stochastische Prozess aus verschiedenen Trajektorien besteht, dessen Realisierung erst im Zeitverlauf enthüllt wird, stellen Erwartungswerte eine gute Möglichkeit dar eine Optimierung vorzunehmen. Nachdem die einzelnen Zustände im Wahrscheinlichkeitsraum voneinander abhängig sind und zukünftige Zustände von vergangenen Ereignissen beeinflusst werden, muss eine Optimierung in der Zukunft beginnen um Erwartungswerte bestimmen zu können und damit die Entscheidungen in der Gegenwart zu beeinflussen.

Will man zum Beispiel den Gewinn eines Produktionsprozesses auf diese Weise optimieren, so muss man zuerst den erwarteten Gewinn der Zustände an dessen Ende bestimmen, um anhand dieser die Entscheidung in der Gegenwart, welche Trajektorie im Endeffekt beschritten wird, zu beeinflussen. Das Mittel um diese rückwärts fortschreitende Optimierung mathematisch auszudrücken, heißt Rekursion. Im mathematischen Sinne ist eine Rekursion eine Berechnungsvorschrift, bei der das Endergebnis von zuvor berechneten Ergebnissen derselben Funktion abhängt. Bei der Berechnung der Fakultät einer Zahl wird dieses Prinzip deutlich.

$$\begin{aligned}n! &= (n - 1)! * n \quad \text{für } n > 1 \\n! &= 1 \quad \quad \quad \text{für } n = 1\end{aligned}$$

**Formel 9: Fakultät**

Diese Berechnungsvorschrift führt die Funktion  $n! = (n - 1)! * n$  solange aus, bis durch Reduktion der Variablen in der Klammer selbige 1 wird. Dadurch entsteht eine lange Reihe von Multiplikationen, welche erst ausgeführt werden kann, wenn das Ergebnis der letzten Klammer berechenbar ist und kein rekursiver Aufruf der Funktion mehr darstellt. Auf diese Art und Weise wird ein großes Problem in viele kleinere und auch trivialere Probleme aufgeteilt, welche dann auch einfach zu lösen sind.

In der Informatik kommt dieses Prinzip in ähnlicher Form zum Einsatz. Auch hier kann ein großes Problem durch die Zerlegung in viele kleine Probleme, welche voneinander durch ihre Lösungen abhängig sind, zerlegt werden. Analog zum mathematischen Berechnungsvorschrift kann in Java eine Methode sich selbst aufrufen, im Falle der Berechnung einer Fakultät ändert sich nur der Übergabeparameter  $n$  dekrementell. Daraus entsteht wie im obigen Beispiel eine Reihe von Multiplikationen, von denen nur die letzte gelöst werden kann, da sie keine Funktion von  $n$  ist. Ist der letzte Wert bestimmt, lässt sich auch der vorletzte Wert ermitteln und so arbeitet sich der rekursive Algorithmus den umgekehrten Weg durch bis zur ersten Multiplikation, die dann das Ergebnis liefert.

Auch in der Informatik heißt die Herangehensweise an obiges Problem Rekursion und ist bei der Realisierung einer SOPPS sowohl in konzeptioneller als auch praktischer Hinsicht von Bedeutung. Hinzu kommt der Vorteil, dass eine rekursive Funktion, welches sich selbst aufruft mit weniger Methodenaufrufen auskommt, als eine iterative Herangehensweise. Das bringt eine deutliche Ersparnis bei der Laufzeit des Algorithmus und viele Probleme der Informatik lassen sich auf diese Art und Weise auch eleganter und einfacher lösen.

In der SOPPS wird das Gesamtproblem einer strategischen Produktionsplanung ebenfalls in mehreren rekursiven Schritten gelöst. Schon bei der Generierung des Wahrscheinlichkeitsraumes in Form einer Baumstruktur kommt eine Rekursion zum Einsatz, da durch eine einfache Wiederholung einer Funktion diese Aufgabe nicht bewältigt werden kann, da die einzelnen Knoten des Baumes ja voneinander abhängig sind. Auch hier werden die Knoten der Baumstruktur bis zur vorgegebenen Periode generiert. Erst dann beginnt der Algorithmus den Baum sozusagen aufzufächern und weitere Kinderknoten zu erzeugen. Dies geschieht solange bis jeder Knoten die vorgegebene Anzahl an Kindern aufweist und der Baum an sich in jedem Ast bis zur vorgegebenen Periode expandiert wurde. Diese zwei Bedingungen stellen sozusagen die Analogie zum besprochenen Beispiel dar. Wenn  $n = 1$  erfüllt ist, hört die Berechnungsvorschrift auf neue Multiplikationen zu generieren.

Dasselbe Prinzip wird auch für die Werte, welche in den Knoten gespeichert werden, angewandt. Werden diese Werte in den Knoten gespeichert, wird der Baum auf dieselbe Art und Weise durchlaufen, wie er generiert wurde. Dies



geschieht mehrmals, da die Werte voneinander abhängig sind. Die Abbruchbedingungen für die Funktion sind freilich nicht dieselben.

Ist der Wahrscheinlichkeitsraum befüllt und wird dieser durchlaufen, so findet eine Informationsenthüllung statt. Wie bereits erläutert bleibt dabei unklar, welcher Zustand am Ende realisiert wird, also muss hier mit Funktionen gearbeitet werden, welche Erwartungswerte über alle noch möglichen Zustände berechnen. Auch das entspricht der rekursiven Denkweise, da wie bei der mathematischen Definition einer Rekursion deutlich wurde, erst am Ende aus einer Variable eine diskrete Größe wird.

Wie genau diese rekursive Denkweise bei SOPPS zum Einsatz kommt, erläutert das nächste Kapitel, welches die Implementierung des Prototyps behandelt. Hier wird dann im Detail erläutert, wie die einzelnen in dieser Arbeit vorgestellten Konzepte zusammenspielen und wie dies im Detail geschieht.

## 3. Implementierung der SOPPS

Das folgende Kapitel befasst sich nun mit dem praktischen Teil dieser Diplomarbeit, einem Prototypen für die modellbasierte Planung und Steuerung unter Unsicherheit. Dabei kommen die vorab diskutierten Konzepte zum Einsatz. Nachfolgend wird der Prozess beschrieben, der zu dem Konzept des Prototyps führte, welche Technologien dabei zum Einsatz kamen und der Prototyp an sich mit einer Beschreibung der Datenstrukturen und Programmlogik. Anhand von in jBPM definierten Prozessen und diversen Screenshots von der Benutzeroberfläche wird abschließend die Bedienung erläutert.

### 3.1. Konzeptionelle Überlegungen

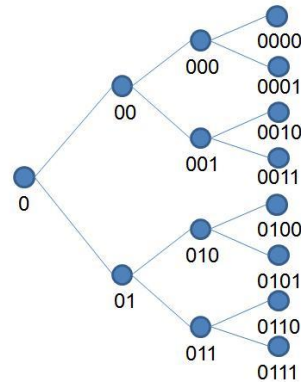
Bevor man sich überhaupt mit einer Implementierung einer SOPPS beschäftigen kann, muss zuerst das zugrundeliegende Modell genau spezifiziert werden sowie die Datenstrukturen, in denen es abgebildet wird. Das fand in einem iterativen Prozess statt, der ein anderes Ende nahm als ursprünglich erwartet.

#### **Baumstruktur:**

Wie bereits beschrieben wird der Produktionsplan mittels  $\sigma$ -Algebra in einer Baumstruktur dargestellt. Eine solche Baumstruktur zu generieren war für den Autor die erste Herausforderung. Eine der vorangegangenen Überlegungen war, nur jene Parameter, die für die Erzeugung des Baumes notwendig sind in die Datenbank zu speichern und den Baum zur Laufzeit zu generieren. Damit spart man sich nicht nur den Speicherplatz für unzählige Knoten in der Datenbank, sondern auch ebenso viele Datenbankzugriffe, was sich natürlich positiv auf die Performanz des Systems auswirkt. Das einfachste Modell eines stochastischen Produktionsplanes ist ein Binärbaum. In diesem Baum hat jeder Knoten zwei Kinder, welche im einfachsten Fall eine gestiegene oder gefallene Nachfrage repräsentieren. Die Baumstruktur dieser Diplomarbeit sollte allerdings mehrere Zustände ermöglichen.

Doch wie viele mögliche Zustände sind sinnvoll, d.h. wie granuliert sollen die einzelnen diskreten Zustände die Realität abbilden? Diese Zahl wird durch die

Vergabe eines ID-Attributs, welche die einzelnen Knoten eines Baumes eindeutig identifiziert begrenzt. Es wurde ein numerischer Schlüssel gewählt, der sich folgendermaßen zusammensetzt. Der für die Schlüsselvergabe zuständige Algorithmus beginnt bei Null zu zählen. Die Wurzel des Baumes hat also die ID 0. Wie in Abbildung 14 ersichtlich setzt sich die ID der Kinder eines Knotens jeweils aus der ID des Elternknotens ergänzt durch eine einstellige fortlaufende Nummer zusammen.



**Abbildung 14: Nummerierung der Baumknoten**

Dieses Nummernsystem hat den Vorteil, dass man aus einer ID heraus direkt die Periode des jeweiligen Knoten bestimmen kann sowie sämtliche Elternknoten und auch Geschwisterknoten. Das wird später bei der Visualisierung des Baumes eine nicht unbedeutende Rolle spielen. Dadurch lassen sich nämlich Teile des Baumes, welche nicht mehr durchlaufen werden können sozusagen wegschneiden, was für den Anwender durch eine farbliche Markierung ersichtlich wird. Wenn wir dabei an den Abschnitt 2.5.2 über die mathematischen Lösungsansätze zurückdenken, wird genau damit die Filtrierung des Wahrscheinlichkeitsraumes visualisiert.

Der gravierende Nachteil dieser Lösung ist, dass die Programmlogik eben mit Hilfe dieses Nummernsystems, wie später erläutert wird, durch den Baum navigiert. Damit man aus einer ID die Bezeichnung eines beliebigen Elternknotens heraus lesen kann, muss man wissen, um wie viele Stellen die ID pro Periode bzw. Generation verlängert wird. Der Algorithmus sieht eine Stelle pro Generation vor, was allerdings die Anzahl der Kinder auf zehn beschränkt, da diese nur von Null bis Neun durchnummeriert werden können. Das warf die Frage auf, wie viele Kinderknoten überhaupt unterstützt werden sollten. Bei der Suche nach einer Lösung des Problems stellte der Autor dabei folgende Schätzung des Speicherbedarfes an. Pro Knoten wurde dabei ein Kilobyte

veranschlagt, was viel zu gering geschätzt ist, allerdings den Speicheraufwand schon dabei in beindruckender Weise veranschaulicht.

<b>Anzahl Kindknoten</b>	<b>Anzahl Knoten gesamt</b>	<b>Speicherbedarf</b>
100	101.010.101	96,3 GB
50	6.377.551	6,1 GB
30	837931	0,8 GB
20	168.421	164 MB
10	11.111	11 MB

**Tabelle 4: Speicherbedarf der Baumstruktur**

Nachdem die Baumstruktur als solche sich im Arbeitsspeicher befindet, wird anhand von Tabelle 4 die Problematik mehrerer Knoten ersichtlich. Der Speicherbedarf steigt bei wachsender Anzahl an Knoten exponentiell an wodurch für die Speicherung eines Baumes, der pro Knoten 100 Kinderknoten hat, ein Arbeitsspeicher mit einer Kapazität von 96,3 Gigabyte notwendig wäre. Diese Zahl bezieht sich wohlgernekt alleine auf den Baum, wobei ein Knoten mit einem Kilobyte veranschlagt wurde. Speicherkapazitäten für die Berechnungen sowie den Betrieb des PCs sowie die Rechenzeit wurden dabei noch gar nicht berücksichtigt. Bei über 100 Millionen Knoten dürfte der Bedarf an Rechenzeit jedoch nicht unbeträchtlich sein. Eine der Rahmenbedingungen bei diesen Überlegungen war auch, dass ERP-Control später auch zu Unterrichtszwecken eingesetzt werden soll. Das heißt, dass die Applikation auch auf einem handelsüblichen Notebook, wie es Studenten verwenden, lauffähig sein muss.

Ein annehmbarer Speicherbedarf ergibt sich erst bei 30 Kinderknoten wobei hier immer noch das Problem auftritt, dass in der Baumstruktur über 800.000 Knoten in einem Browserfenster dargestellt werden sollen. Das macht die Applikation sehr unübersichtlich und wirft das bekannte Problem mit der Nummerierung der Knoten auf. Aus diesen Überlegungen heraus wurde die Anzahl der möglichen Kinderknoten auf Zehn beschränkt. Das löst das Nummerierungsproblem und der Wahrscheinlichkeitsraum als Modell der Realität weist mit zehn möglichen Zuständen, die am Ende einer Periode realisiert werden können, auch einen ausreichend hohen Detaillierungsgrad auf.

### Ereignisse und Entscheidungen:

Ein weiterer Knackpunkt bei der Modellbildung der SOPPS war die Belegung der Knoten mit Werten. In einer Planungsperiode kann die Marktsituation verschiedene Nachfragemengen als Zustand annehmen. Ziel der SOPPS ist es ja, die optimale Produktionsmenge zu finden, um die Nachfragemenge möglichst vollständig zu befriedigen und die Lagerkosten zu minimieren. Das führte dazu, dass zu einer bestimmten Marktsituation zusätzliche Zustände durch die Produktionsmenge dazu kamen, welche ja den Lagerstand beeinflussen. Um das zu veranschaulichen kehren wir zu dem Binärbaum als Beispiel zurück. Nur in diesem Fall haben wir sozusagen zwei Bäume. Die folgende Abbildung 15: Problem multipler Zustandsvariablen soll das Dilemma besser illustrieren.

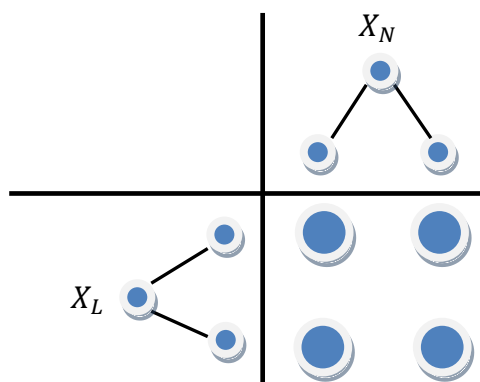


Abbildung 15: Problem multipler Zustandsvariablen

Wenn in einem Binärbaum die nächste Periode zwei Zustände annehmen kann, also entweder steigende oder fallende Nachfrage, dann kann man auch nur auf zwei Arten die Produktionsmenge daran anpassen, entweder man steigert oder drosselt die Produktion. Daraus ergeben sich allerdings nach Durchlaufen der betrachteten Periode vier Zustände. Geht man von einer steigenden Nachfrage aus und tritt dies nicht ein steigt die Lagermenge. Liegt man richtig, steigt die Lagermenge nicht. Demselben Schema folgt diese Überlegung bei gesunkener Nachfrage. Daraus stieg die Komplexität des Modells. Um diesem Problem zu begegnen wurden zwei Arten von Knoten in einer früheren Version des Baumes eingefügt. Eine Sorte bildete dabei Entscheidungen ab, also musste man sich hier entscheiden, welche Nachfragemenge für die kommende Periode annimmt. Die zweite Sorte waren Ereignisknoten, welche dann das Ergebnis darstellten.

In der entgültigen Implementierung wurde diesem Problem Rechnung getragen, indem die optimale Produktionsmenge für eine Periode bereits in dem Knoten der

Vorperiode gespeichert wird. Dadurch wird die Komplexität multivariabler Zustände in der Darstellung entschärft, da sich z.B. in einem binomialen Modell jeweils nur einer von zwei Zuständen realisieren kann. Die Entscheidung der Produktionsmenge erfolgt bereits in der Vorperiode.

### **Denkfehler stochastischer Prozess:**

Trotz häufiger Diskussionen und Treffen zwischen dem betreuenden Professor und dem Autor schlichen sich Denkfehler beim Verständnis der stochastischen Konzepte ein. Erst nachdem die Baumstruktur klar definiert und ausgearbeitet war, wurden diese Denkfehler offensichtlich. Das Ziel des Autors war es, den optimalen Pfad innerhalb der Baumstruktur zu bestimmen. Die Schwierigkeit dabei waren die Wahrscheinlichkeiten der einzelnen Zustände. Zur Vereinfachung des Modells wurden alle Zustände innerhalb einer Periode als gleich wahrscheinlich angenommen. Wie sollte man aus diesen einen Zustand auswählen? Als Entscheidungskriterium wurde ja die Umsatzergebnisfunktion herangezogen, welche für den sehr optimistischen aber unwahrscheinlichen Fall einer stetig wachsenden Nachfrage die besten Ergebnisse liefert. Das entscheidende Stichwort dabei war Funktion. Anstatt in Werten zu denken und eine optimale Produktionsstrategie aus konkreten Zahlen zu berechnen, lag die Lösung in einem funktionsorientierten Denkansatz.

Die Strategie wird in der jetzigen Lösung nicht durch einen Pfad innerhalb des Planungsbaumes definiert, sondern spiegelt sich bereits in der zugrundeliegenden Spezifikation des Baumes wieder. Hier werden ja sämtliche Parameter gespeichert, um den Baum zu generieren und mit Werten zu befüllen. Wird dieser strategische Produktionsplan von der Geschäftsleitung oder einer anderen zur Entscheidung ermächtigten Instanz genehmigt, wird dieser ja auch ausgeführt und es werden Commitments daraus generiert, welche ja Aufträge für die Produktion darstellen, die sich dann mit der Feinplanung beschäftigt um diesen betriebsinternen Auftrag zu erfüllen. Wenn wir uns an die Begriffsdefinition erinnern eines strategischen Plans erinnern, dann legt dieser ja Ziele fest und nicht den Weg um diese zu erreichen. Der Plan repräsentiert also keinen vordefinierten Pfad innerhalb des Baumes sondern stellt vielmehr ein Modell dar, nach dem die Produktion agieren soll. Nach jeder vollendeten Periode werden so die verbleibenden Perioden des Planes neu bewertet und damit können sich auch die beschrittenen Pfade des Baumes mit jeder Periode ändern. Dabei wird auch die Filtrierung der Wahrscheinlichkeitsräume ausgenutzt, da im Zeitverlauf

immer weniger Zustände realisierbar sind und damit die Vorhersage anhand von Erwartungswerten des Betriebsergebnisses immer genauer wird.

Diese Erkenntnisse hatten auch Auswirkungen auf die Baumstruktur. In der finalen Version tritt die Problematik multipler Zustandsvariablen gar nicht erst auf, da ja nicht mehr auf der Suche nach einem optimalen Pfad anhand von Erwartungswerten die Produktionsmenge kalibriert wird, sondern der Plan den Handlungsspielraum vorgibt, innerhalb dessen sich der Anwender respektive die Produktionsabteilung bewegen kann. Die Produktion kann nur innerhalb der vorab definierten Grenzen operieren und daher sind die Lagerstände ebenfalls im Vorfeld bekannt, nur eben nicht, welchen Zustand innerhalb des Baumes ausgeführt wird. Damit wurden auch die unterschiedlichen Arten von Knoten überflüssig und eine Sorte von Knoten ist ausreichend.

## **3.2. Verwendete Technologien**

Der folgende Abschnitt beschäftigt sich mit den wichtigsten für die Implementierung des Prototyps verwendeten Technologien. Dabei wird weniger auf technische Details eingegangen. Vielmehr sollen die einzelnen Technologien und Frameworks kurz vorgestellt werden um dem Leser einen Eindruck zu vermitteln, was sie leisten und warum sie für die Implementierung sinnvoll sind. Die Wahl der Technologien oblag dabei nicht dem Autor, sondern war schon durch einen Prototyp von ERP-Control vorgegeben, der eben durch SOPPS im Rahmen dieser Diplomarbeit erweitert wurde.

### **3.2.1. Enterprise Java Beans**

Enterprise Java Beans (EJB) sind ein integraler Bestandteil des Java Enterprise Edition (J2EE/JavaEE). Im Unterschied zur Standardversion von Java liegt der Fokus bei Java EE auf webbasierte Anwendungen im Unternehmensumfeld. EJBs stellen dabei Softwarekomponenten dar. Hier wird zwischen zwei Arten unterschieden. Session Beans beinhalten die Geschäftslogik und unterteilen sich wiederum in zustandsbehafteten und zustandslosen Session Beans. Entity Beans repräsentieren die Geschäftsobjekte, auf denen die Session Beans operieren.

Message-Driven Beans fungieren als Kommunikationsmittel zwischen Client und Server.

#### **Stateless Session Bean (SLSB):**

SLSBs dienen als Container für einfache Operationen am Server. Dies können z.B. einfache Berechnungen oder der Aufruf weiterer EJBs sein. Namensgebend speichert eine SLSB keine Zustände. Nach Beendigung ihrer Aufgabe wird die jeweilige Instanz der SLSB gelöscht. Das heißt also, dass ein Client, wenn er wiederholt während einer Sitzung bzw. neudeutsch Session dieselbe SLSB aufruft, nicht zwangsläufig dieselbe Instanz dieser Session Bean erhält und damit mitunter auch nicht dieselben Daten wie beim vorherigen Versuch. Vgl.[28].

#### **Stateful Session Bean (SFSB):**

Im Gegensatz dazu speichert die SFSB sehr wohl Zustände innerhalb einer Session. Diese wird nämlich direkt einem Client zugeordnet und existiert anders als die SLSB bis zu einer expliziten Löschung oder langer Inaktivität. Der erlaubte Zeitraum der Inaktivität wird hierbei von der Applikationslogik geregelt. Beispielsweise könnte das eine automatische Abmeldung eines Anwenders sein, wenn dessen Client zu lange inaktiv ist. Ein Session Bean könnte beispielsweise einen Geschäftsprozess implementieren und dessen Zustände speichern, in denen sich der Client gerade befindet.

#### **Entity Bean:**

Wie bereits erwähnt repräsentieren Entity Beans die Geschäftsobjekte innerhalb einer Applikation. In der Literatur werden diese auch als Plain Old Java Objects (POJO) bezeichnet, also als einfaches Objekt, das bis auf Zugriffsmethoden auf seine Attribute ohne spezielle Funktionalität auskommt. Die Geschäftslogik wird ja durch die Session Beans gekapselt. In den Entity Beans finden sich jedoch Annotationen, die für die Speicherung des Objektes in der Datenbank benötigt werden.

#### **Message-Driven Bean (MDB):**

Der Vollständigkeit halber sei auch noch die Message-Driven Bean an dieser Stelle angeführt. Diese werden zur asynchronen Kommunikation zwischen Client und Server verwendet. Dies ist dann der Fall, wenn Anfrage und Antwort zeitlich deutlich versetzt stattfinden, weil z.B. das Senden einer Antwort erst durch einen Anwender bestätigt werden muss. Diese Art von EJB findet bei ERP-Control während der Erstellung dieser Diplomarbeit jedoch keine Anwendung.



### 3.2.2. Hibernate

Nachdem die meisten Literaturquellen zu diesem Thema eine nahezu identische Einleitung haben, soll auch hier dieses Thema zur Sprache kommen. Das Konzept einer relationalen Datenbank bedient sich ja bekanntermaßen einer tabellarischen Struktur. Jede Zeile einer Tabelle soll dabei durch einen Schlüssel, der durchaus auch aus Werten mehrerer Spalten bestehen kann, eindeutig identifiziert werden. Zur Verknüpfung einzelner Tabellen miteinander werden Fremdschlüssel eingesetzt. Diese zeigen an, dass die Daten aus Zeilen mehrerer Tabellen zueinander gehören. Dieser Fremdschlüssel findet sich entweder in einer der betroffenen Tabellen oder auch in einer eigenen Tabelle, welche dann nur Schlüssel enthält. Soviel zu einem relationalen Datenbankschema.

In der objektorientierten Programmierung kommen allerdings andere Konzepte zum Einsatz. Wie der Name schon sagt wird hier mit Objekten gearbeitet. Die Rede ist von POJOs oder im Kontext von J2EE auch Session Beans, welche aus bestimmten Attributen bestehen, wobei ein Attribut wiederum ein Objekt sein kann, das sich aus mehreren Attributen zusammensetzt. Will man ein derart komplex strukturiertes Objekt in eine relationale Datenbank speichern, wird das Dilemma offensichtlich. Attribute mit einfachen Datentypen wie Zahlen oder Buchstaben lassen sich einfach in Spalten einer Tabelle abspeichern. Ist das Attribut allerdings selbst wieder ein Objekt, das über mehrere Attribute verfügt, wird die Persistierung eines solchen Objektes schon schwieriger. Die Konzepte zur Datenspeicherung zwischen relationalen Datenbanken und objektorientierten Programmiersprachen unterscheiden sich also grundlegend. Dieses Dilemma wurde in den 90er-Jahren des vergangenen Jahrhunderts unter dem Namen Object-relational Impedance Mismatch bekannt.

Konzepte, um diesem Problem beizukommen werden unter dem Oberbegriff object-relational mapping (ORM) zusammengefasst. Anfangs mussten Programmierer sich noch selbst um ein Mapping kümmern, sprich Objekte vor der Speicherung in einer Datenbank in eine entsprechende Darstellungsform bringen und bei komplexeren Objekten sich ebenfalls um die Verteilung der Daten auf mehrere Tabellen zu kümmern. Der Aufwand für jede Applikation eine eigene Logik zur Persistierung der Daten zu schreiben sowie deren Fehleranfälligkeit schrie förmlich nach einer Vereinfachung der Programmierarbeit durch ein Framework. Eines dieser Frameworks ist Hibernate, welches die häufigsten Aufgaben bei der Persistenz von Objekten übernehmen

soll. Der Programmierer muss keine Logik mehr schreiben, sondern bekommt fertig implementierte Funktionen zu Verfügung gestellt, die sich um die Persistenz kümmern.

In welcher Form ein Objekt in eine relationale Datenbank gespeichert werden soll, wird in Hibernate durch eine XML-Datei beschrieben. Diese Mapping-Datei stellt eine Konfiguration dar, wie Hibernate einzelne Objekte speichern soll. Dabei werden sowohl Mappings zwischen Attributen eines Objekts und Spalten einer Tabelle abgedeckt als auch die Beziehungen von Objekten untereinander und deren Repräsentation in der Datenbank.

In Verbindung mit der Java Persistence API (JPA), welche Teil von Java EE ist, und entsprechenden Tools, die man bei Entwicklungsumgebungen wie Eclipse einbinden kann, kann man dem Programmierer die Konfiguration mittels XML-Dateien ersparen. Bei JPA werden Annotations verwendet, um eine kontextbasierte Zuordnung der Bedeutung von Objekten und deren Attributen zu realisieren. Dadurch kann schon im Code spezifiziert werden, welche Objekte in die Datenbank gespeichert werden sollen, wie dessen Attribute für die Persistenz in der relationalen Datenbank konvertiert werden sollen oder welche Attribute den Schlüssel bilden. Dadurch kann aus diesen Annotations heraus die Konfigurationsdatei für Hibernate generiert werden.

### **3.2.3. JavaServer Faces**

Die Hypertext Markup Language oder bekannter unter dem Akronym HTML war ursprünglich dazu bestimmt, Text auf Webseiten zu formatieren. Indem ein Text von verschiedenen Tags eingeschlossen wird, kann man unterschiedliche Formatierungen, Aufzählungen, Tabellen udgl. realisieren. Will man Java-Applikationen mit einem Webinterface ausstatten, so wurde der HTML-Code früher durch Methoden generiert. Es entstand eine bunte Mischung aus Applikationslogik und Darstellungslogik, was wegen der schlechten Wartbarkeit ganz und gar nicht im Sinne des MVC-Gedankens steht. Ein weiteres Manko bestand darin, dass HTML im Java-Code in Form von Strings abgelegt war, welche vom Compiler natürlich nicht auf syntaktische Korrektheit überprüft wurden. Viele Fehler, im HTML-Code wurden so erst zur Laufzeit entdeckt, was die Entwicklung einer Webapplikation natürlich verlangsamte.

Die logische Konsequenz war eine Technologie zu schaffen, die diese bunte Mischung wieder sauber trennte – JavaServer Pages (JSP) war das Ergebnis dieses Prozesses. Bei JSP wurde konventioneller HTML-Code ergänzt durch zusätzlichen Tag, welche sogenannte Scriptlets beinhalteten. In diesen Skriptlets wurden die Methodenaufrufe platziert. Dadurch wurden anfangs wieder Logik und Präsentation sauber getrennt. Die Idee war an sich gut, allerdings schlugen die Programmierer eine andere Marschrichtung ein als von den Schöpfern von JSP vorgesehen. JSP hilft dabei das statische Gerüst von HTML-Seiten zu erstellen und mit Werten zu befüllen. Das Problem ist die Dynamik. Je nach Zustand der Webapplikation sollen Teile einer Seite anders gestaltet werden können, was natürlich auch einen unterschiedlichen HTML-Code impliziert. Vgl. [29]. Anstatt HTML-Code in Java-Code zu integrieren passierte nun das genaue Gegenteil. Java-Methoden wurden in HTML-Code integriert und am Ende dieser Entwicklung entstanden wieder bunte „Code-Mischungen“, die eben so schlecht zu warten waren wie die vorangegangenen Lösungen. Desweiteren litt die Wiederverwendung von Code unter eben dieser Mischung.

JavaServer Faces (JSF) sollte eben genau diese Problematik beheben. Die Hauptvorteile von JSF liegen in der Wiederverwendbarkeit von graphischen Komponenten durch eine konzeptionelle Strukturierung des Inhalts einer Webseite mittels Vorlagen (Templates) und der Steuerung von Systemzuständen während der Benutzung. In diesem Zusammenhang sind auch Kompositkomponenten von Interesse. Das sind im Prinzip Seitenfragmente, welche in XHTML definiert werden. Aus diesen Fragmenten lassen sich dann in weiterer Folge komplette Webseiten zusammensetzen. Ein Beispiel für so eine Kompositkomponente wäre z.B. eine Menüleiste, welche zur Navigation in einem Internetauftritt dient und sich meistens als Linksammlung am linken Bildschirmrand darstellt. Diese Menüleiste ist auf jeder Seite eines Internetauftritts gleich und wird so einmal definiert und immer wieder verwendet. Eine der Hauptbestandteile von JSF sind vordefinierte Klassen für das User Interface (UI). Für die gängigsten Aufgaben des UI wie z.B. Eingabe, Ausgabe oder Darstellung werden Default-Implementierungen bereitgestellt.

Mittlerweile gibt es diverse Erweiterungen zu JSF, das dieses Framework um einige nützliche Komponenten erweitert. Gerade in Zeiten wo Web 2.0 ein nahezu omnipräsentes Schlagwort in der Webentwicklung ist spielen dynamische Websites eine immer größere Rolle. Ein Konzept, das diese Entwicklung unterstützt ist Ajax. Das ist ein Akronym für Asynchronous JavaScript And XML

und beschreibt keine Technologie oder Framework, sondern vielmehr eine Sammlung von Technologien zur Realisierung von asynchroner Datenübertragung, allen voran die Skriptsprache JavaScript, welche Programmlogik in Webseiten einbaut. Dadurch werden Anfragen an den Server möglich, die ausgeführt und dargestellt werden können ohne eine Website komplett neu zu laden. Das ermöglicht eine im Sinne von Web 2.0 dynamischere Interaktion zwischen Server und Client sowie auch zwischen Client und Anwender, da die zu übertragene Datenmenge verringert wird und die Bedienung einer solchen Webseite flüssiger abläuft. Vgl.[30]. Erweiterungen von JSF, die Ajax unterstützen sind ergänzende Bibliotheken zum JSF-Framework wie z.B. RichFaces oder auch ICEFaces.

### **3.2.4. JBoss Seam**

JBoss Seam ist ein Framework, welches auf Java Enterprise Edition aufsetzt. Wie der Name schon vermuten lässt, soll es eine Naht sein. Aber wozu dient diese Naht, was verbindet sie? Java EE an sich ist schon eine Ansammlung von Frameworks. Seam ist „das fehlende Framework, das Bestandteil von Java EE [...] hätte sein sollen.“[31] Konzeptionell ist es über Java EE positioniert und soll es dem Programmierer durch ein einfach zu handhabendes Programmiermodell erleichtern, sämtliche Komponenten einer Enterprise-Webanwendung zu implementieren. Ziel dahinter soll es nicht noch sein, dem Programmierer die Arbeit zu erleichtern, sondern dadurch auch eine gesteigerte Effektivität zu erreichen und die Anwendungen besser skalieren zu können.

Wie bereits angedeutet ist Java EE selber eine Ansammlung von Frameworks. Dessen Kernkomponenten sind Enterprise Java Beans (EJB) sowie JavaServer Faces (JSF). EJBs beinhalten hierbei die Geschäftslogik während JSF für die Darstellung zuständig ist. Diese Aufteilung ist weit verbreitet und nennt sich Model-View-Controller-Muster (MVC). Dabei werden das Datenmodell, die Präsentation sowie die Programmlogik voneinander getrennt. Das erleichtert nicht nur die Programmierung, sondern erleichtert auch die Wartung und Erweiterung bestehender Software. Aus diesen Überlegungen heraus ergänzen sich JSF und EJB ausgezeichnet, trotzdem sind sie zwei unterschiedliche Frameworks, die auch unterschiedlichen Konzepten zugrunde liegen. Während JSF XML-Dateien zur Konfiguration verwendet, geschieht dies bei EJBs durch Annotationen direkt im Programmcode. Diese Unterschiede erschweren die

Zusammenarbeit dieser beiden Rahmenwerke und genau an diesem Punkt setzt Seam sozusagen Nadel und Faden an. Anstatt wiederum eigene Konzepte einzuführen um JSF und EJB zu verbinden (z.B. JSF Backing Beans), wird dieser Aufwand durch Seam überflüssig. Wie EJB verwendet Seam auch Annotations, um ohne Umwege beide Frameworks gemeinsam verwenden zu können.

Doch Seam ist nicht nur ein Flickwerk, welches unterschiedliche Technologien unter einem Dach zusammenführt, es kann noch mehr. Was an dieser Stelle nicht unerwähnt bleiben darf, ist die Unterstützung der ebenfalls von JBoss stammenden Geschäftsprozess-Engine jBPM. Dieses Werkzeug erlaubt es dem Programmierer, GP graphisch zu spezifizieren und mit Hilfe dieser Definition die Abläufe der Applikation zu administrieren. Hinter der graphischen Repräsentation eines GP liegt eine XML-Datei, in welcher die einzelnen Knoten und Übergänge des GP beschrieben werden. In diese Schablone muss im Anschluss an die Definition nur noch die jeweilige Java-Klasse eingefügt werden, die für Realisierung des GP sorgen. Dadurch wird es auch möglich, dass eine Fachabteilung direkt an der Softwareentwicklung beteiligt ist auf diese Art die eigenen GP definiert, die der Programmierer dann direkt weiter verwenden kann.

Seam ist daher der gemeinsame Nenner, der die zuvor vorgestellten Technologien miteinander verbindet.

### **3.3. Prototyp der SOPPS**

Dieser Abschnitt stellt nun die Lösung der Forschungsfrage dieser Arbeit in Form einer Implementierung eines Prototyps für die SOPPS vor. Mit den in dieser Arbeit vorgestellten Konzepten, Standards, Normen und Denkweisen als Rüstzeug wurde die Implementierung basierend auf einem Modell mit solidem theoretischem Fundament erstellt. Zu aller Erst widmet sich dieser Abschnitt der zentralen Datenstruktur dieses Prototyps, einer Baumstruktur, welche in Breite und Tiefe flexibel gehalten wurde, um dem späteren Anwender so viele Freiheiten wie möglich bei der Planung offen zu lassen. Im Anschluss wird erläutert, wie diese Baumstruktur mit Werten befüllt wird und wie die kybernetische Steuerung auf Grundlage dieses Baumes funktioniert. Abschließend werden die einzelnen Funktionalitäten anhand von Use Case-Diagrammen und Prozessen beschrieben sowie die Benutzeroberfläche in Auszügen vorgestellt.

### 3.3.1. Datenstrukturen

Bevor damit begonnen werden kann einen Baum und damit die SOPPS zu implementieren müssen zwei Datenstrukturen definiert werden, welche zur dauerhaften und flüchtigen Speicherung von Daten verwendet werden. Diese werden nachfolgend kurz vorgestellt.

#### 3.3.1.1. ProductionPlanModelEntity-Klasse

Die Klasse ProductionPlanModelEntity ist, wie die Annotation `@Entity` in der ersten Zeile von Codefragment 1 anzeigt ein Entity Bean und wird in der Datenbank gespeichert. Diese Klasse beinhaltet sämtliche Parameter, welche für die Generierung des Planungsbaumes notwendig sind und verfügt ansonsten über keine höhere Funktionalität. Diese Parameter werden nachfolgend kurz beschrieben.

```
@Entity
@DiscriminatorValue("PRODUCTION")
public class ProductionPlanModelEntity extends PlanModelEntity {

    private String product_ID; // to refer to a product
    private Date startDate;
    private int periods; // depth of the tree
    private int cardinality; // no. of children of each tree node
    private Double[] spreadOfDemandPerPeriod; // percentage of spread
                                                of assumed demand

    private Double overallDemand;
    private Double[] percentageOfDemandPerPeriod;
    private Double startStorage;
    private Double minProduction;
    private Double maxProduction;
    private Double productPrice;
    private Double productionStep; // amount, which the production
                                    output can be increased

    private Double fixedCosts;
    private Double variableProductionCosts;
    private Double variableStorageCosts;
    private String currentNodeID; // saves the current Node in
                                    PlanningTree

    [...]
}
```

Codefragment 1: Klasse ProductionPlanModelEntity

- **product\_ID**: ID des zu produzierenden Produktes
- **startDate**: Startzeitpunkt der Produktion
- **periods**: Dauer der Produktion in Quartalen
- **cardinality**: Anzahl der Kindknoten pro Knoten
- **spreadOfDemandPerPeriod**: Schwankungsbreite der Nachfrage für jedes Quartal des Jahres

- **overallDemand**: Gesamtmenge, die über die Dauer der Produktion hergestellt werden soll
- **percentageOfDemandPerPeriod**: Anteil des Gesamtbedarfes pro Periode
- **startStorage**: Lagerstand, der zu Beginn der Produktion vorhanden ist
- **productPrice**: jener Preis pro Einheit, um den das Produkt an den Kunden verkauft wird
- **minProduction**: minimale Ausbringungsmenge, die pro Periode produziert werden kann
- **maxProduction**: maximale Ausbringungsmenge, die produziert werden kann
- **productionStep**: Menge, um die die Produktion gesteigert bzw. verringert werden kann.
- **fixedCosts**: Fixkosten
- **variableProductionCosts**: variable Produktionskosten
- **variableStorageCosts**: variable Lagerkosten
- **currentNodeID**: ID des Zustandes, in dem sich der Produktionsplan aktuell befindet

Die Variablen **periods** und **cardinality** bestimmen dabei das Aussehen des Planungsbaumes. Die Perioden geben dabei die Tiefe des Baumes an und die Kardinalität eben die Anzahl der Kinderknoten, also wie fein der Plan gefächert ist oder mit anderen Worten wie viele unterschiedliche Möglichkeiten einer zukünftigen Entwicklung im Plan abgebildet sind.

Ebenfalls von besonderen Interesse sind die Variablen **minProduction**, **maxProduction** und **productionStep**. Sie bilden technologische Grenzen bei der Produktion ab, da jedes Produktionssystem und damit auch die Ausbringungsmenge gewissen Grenzen nach oben und unten unterworfen ist. Nachdem Ausbringungsmengen unter Umständen nicht stetig steuerbar sind wird auch dieser Umstand bei der späteren Berechnung der optimalen Produktionsmenge berücksichtigt.

### 3.3.1.2. Node-Klasse

```
public class Node {
    private String id;
    private int period;//useful for building the tree
    private int childCounter; //useful for filling the tree with values
    private boolean possible; //to show if a node realizeable or not
    private Double demand; //saves the planned demand
    private Double expectedDemand;//saves expected demand of next
        period, used for evaluating best production strategy
    private Double supply;//saves the optimal production output, same
        value for every sibling node

    private Double storage;
    private Double profit;
    private Double totalExpectedProfit;

    /*CONSTRUCTOR*/
    /**
     * @param plan_ID
     * @param id
     * @param period
     */
    protected Node(String id, int childCounter, int period) {
        [...]
    }

    /*GETTER-/SETTER-METHODS*/
    [...]
}
```

Codefragment 2: Klasse Node

Die Klasse Node dient zur Speicherung von Werten im Planungsbaum. Im Gegensatz zur ProductionPlanModelEntity wird diese Datenstruktur nicht in der Datenbank gespeichert. Unter Verwendung von Daten aus der ProductionPlanModelEntity kann der Planungsbaum bei Bedarf jederzeit neu im Arbeitsspeicher generiert werden. Folgende Daten werden in dieser Datenstruktur gespeichert:

- **id**: ID des Knotens
- **period**: Periode des Knotens
- **childCounter**: das wievielte Kind des Elternknotens dieser Knoten darstellt. Diese Variable ist bei der Befüllung des Baumes mit Werten hilfreich.
- **possible**: kennzeichnet, ob sich der Knoten auf einem Pfad befindet, welcher beschränkt wurde oder beschränkt werden kann oder durch die Filterung des Wahrscheinlichkeitsraumes ausscheidet.
- **demand**: gibt die Nachfrage an, die befriedigt werden soll, wenn sich die Produktion in diesem Zustand (Knoten) befindet.
- **expectedDemand**: gibt die erwartete Nachfrage an.
- **supply**: gibt die optimale Produktionsmenge für diesen Zustand an.
- **storage**: speichert eventuell vorhandene Lagermengen
- **profit**: speichert das Umsatzergebnis auf Basis einer optimalen Produktion



- **totalExpectedProfit:** gibt das erwartete Umsatzergebnis über alle verbleibenden Perioden an. Diese Variable findet Verwendung bei der intertemporalen Optimierung und stellt einen kumulierten Barwert des Umsatzergebnisses über die verbleibenden Perioden der noch möglichen Trajektorien bedingt auf den jeweiligen Zustand dar.

Ansonsten verfügt diese Klasse über keine nennenswerten Besonderheiten und erfüllt sämtliche Merkmale eines POJOs, da hier nur Daten gespeichert werden und die Klasse über keinerlei Funktionalität verfügt.

### **3.3.2. Auszüge der Geschäftslogik**

Der folgende Abschnitt erläutert in Auszügen die zugrundeliegende Geschäftslogik, nach der das SOPPS-Modul von ERP-Control arbeitet. Es werden die wesentlichen Konzepte erläutert, um dem Leser die Denkweise und Überlegungen, welche zu den nachfolgenden Ergebnissen führte, näher zu bringen. Dabei handelt es sich wie die Überschrift schon aussagt um Auszüge. Triviale Berechnungen, welche durch Hilfsfunktionen durchgeführt werden, bleiben an dieser Stelle unerwähnt.

#### **3.3.2.1. Generierung der Baumstruktur**

Am Anfang eines jeden Baumes steht die Wurzel. Diese stellt einen Sonderfall dar, da sie weder einen Elternknoten noch Geschwisterknoten hat. Aus diesem Grund wird sie auch separat generiert. Der Baum als solches ist zur Initialisierung leer, daher werden die Knoten erst in einem späteren Arbeitsschritt mit Werten befüllt.

Codefragment 3 zeigt den Code zur Erstellung des Baumes, welcher im Anschluss im Detail erläutert wird. Die Wurzel (root) dient nun als Ausgangspunkt für die eigentliche Generierung des Baumes. Zur Speicherung des Baumes wird die Datenstruktur Map verwendet. Nachdem die Konzeption der Datenstruktur zu den ersten Schritten bei der Implementierung eines Systems gehört, wurde Map gewählt, da es sich hierbei um ein Interface handelt und den Code generisch hält. Im Endeffekt wurde die Datenstruktur TreeMap gewählt, um die Knoten des Baumes zu speichern.

```

protected void makeRoot(Double startStorage) {
    Node root = new Node("0",0,0);
    root.setStorage(startStorage);
    planNodes.put("0", root);
}

/**
 * method creates all children of the root node and their IDs
 * needed for navigation in the tree.
 */
protected void makeChildren (String plan_ID, String parent_ID,
                             int period){
    int childCounter = 0;
    do{
        String node_ID=
            parent_ID.concat(String.valueOf(childCounter)); //0->00
        Node cn = new Node (node_ID,childCounter,period);
        planNodes.put(cn.getId(), cn);

        if (cn.getPeriod() < periods){ //if max. period of current
            tree path, reached...
            period++;
            makeChildren(plan_ID, parent_ID, period);
        }
        childCounter++;
    }
    while(childCounter <= cardinality);
}

```

**Codefragment 3: Code zur Generierung des Baumes**

Diese trägt hier den Namen `planNodes` und kann beliebige Objekte speichern, welche durch einen Schlüssel adressiert werden können. Im konkreten Falle der SOPPS dient die vorhin besprochene Nummerierung der Knoten. Bei der Methode `makeChildren` handelt es sich um eine Rekursion, also einer Funktion, die sich selbst erneut aufruft und dabei eine neue Instanz ihrer selbst erzeugt. Dieses Konzept bei der Programmierung ist sehr hilfreich, wenn es um funktionsbasierte Anwendungen geht wie z.B. die Fakultät einer Zahl zu ermitteln. Mit Hilfe der Zählervariablen `childCounter` wird bei der Generierung eines Knotens bestimmt, wie viele Geschwister bereits generiert wurden. Daraus ergibt sich in Verbindung mit der ID des Elternknotens die ID des Kindknotens. Der Knoten wird sogleich erstellt.

Mit Hilfe der Variablen `periods`, welche den Planungshorizont, also die Anzahl der geplanten Perioden repräsentiert, und der Variable `period`, die in jedem Knoten gespeichert ist, überprüft der Algorithmus, ob der Planungshorizont schon erreicht ist oder nicht. Falls dies nicht der Fall ist, wird rekursiv eine neue Instanz der Methode aufgerufen, bis der Planungshorizont erreicht ist.

Bis jetzt hat der Algorithmus mehrere Instanzen der Methode `makeChildren` erzeugt sowie in jeder Periode des Planes einen Knoten. Nun werden mit Hilfe einer Schleife die Geschwister der jeweiligen Knoten erzeugt. Dies geschieht

solange, bis `childCounter` die Anzahl von `cardinality` erreicht hat. Diese Variable wurde gleich wie `periods` vom Anwender bei der Erstellung des Planes angegeben und gibt an, wie viele Kinder jeder Knoten hat, also wie detailliert der Ereignisraum abgebildet wird. Diese Prozedur wiederholt sich solange, bis jede rekursiv aufgerufene Instanz von `makeChildren` diese Bedingung erfüllt hat. Abbildung 16 soll dieses Prinzip veranschaulichen und zeigt die Reihenfolge, in der die Knoten eines binären Baumes generiert werden.

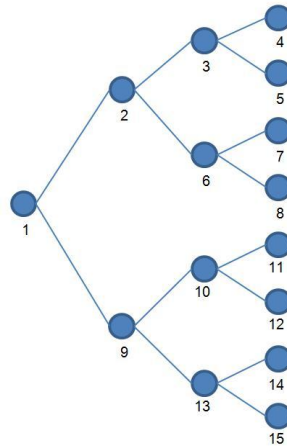


Abbildung 16: Ablauf der Baumgenerierung

Nach diesem Strickmuster werden auch alle anderen Aufgaben, welche die Befüllung des Baumes einzelner Teilbäume betreffen durchgeführt. Dabei ist der Algorithmus schon von Vorn Herein darauf ausgelegt, Teilbäume zu bearbeiten. Als Übergabeparameter findet man immer jeweils eine Knoten-ID sowie die Periode des jeweiligen Knotens wieder. Das ist der Ausgangspunkt für den Algorithmus, von wo aus der Baum bzw. Teilbaum durchlaufen wird.

### 3.3.2.2. Kalibrierung des auf die Baumstruktur aufgesetzten stochastischen Prozess

Exemplarisch wird hier der erste Schritt der Befüllung des Baumes mit Werten erläutert. Dies geschieht wie bereits erwähnt nach demselben Muster, wie die Generierung des Baumes. Der einzige Unterschied ist, dass Knoten nicht neu in `planNodes` eingelesen werden, sondern ausgelesen und mit Werten befüllt überschrieben werden. Um die Komplexität des Codes niedrig zu halten, wurde eine Reihe von Hilfsmethoden implementiert, die einfache Aufgaben lösen. Die einzelnen Knoten werden in mehreren Schritten mit Werten belegt, da diese zum

Teil voneinander abhängig sind und nicht in einem Durchgang berechnet werden können. Schritt 1 dabei ist die Berechnung von einigen Werten, die zur Befüllung von **demand** aus der Klasse Node notwendig sind. Die Belegung dieser Variable mit Werten wird hier stellvertretend für die Befüllung der restlichen Variablen erläutert. Die Aufteilung in mehrere Schritte ist notwendig, da der Anwender ja die Wahl haben soll, das zugrundeliegende Modell entweder selbst zu kalibrieren oder aus historischen Daten eine Kalibrierung berechnen zu lassen.

Dabei hilft die Methode **ContingentPerPeriod**, welche aus der geplanten Gesamtproduktionsmenge und der prozentuellen Anteile der einzelnen Perioden die Nachfrage in jedem Knoten berechnet. Dieser Anteil kann bei der Erstellung des Modells entweder vom Anwender angegeben werden oder aus historischen Werten kalibriert werden. Diese Werte stammen aus dem Objekt **plan** der Klasse **ProductionPlanModelEntity**, welche der Methode als Parameter übergeben wird und die Parameter zur Generierung des Baumes speichert. Weiters wird aus diesen Werten die Variationskoeffizient der Nachfrage pro Periode (**getSpreadOfDemandPerPeriod()** [np]) ermittelt sowie die Schrittweite der einzelnen Nachfragewerte für jeden Knoten der Periode (**DeviationOfDemandPerPeriod**[np]). Nachdem sich beim ersten lauffähigen Prototypen herausstellte, dass es bedingt durch die binäre Speicherung von Dezimalzahlen zu unschönen Zahlen mit periodischen Kommastellen kam, wurde wie in Codefragment 4 ersichtlich bei der Berechnung der Nachfrage auf die Datenstruktur **BigDecimal** zurückgegriffen, welche auch im Dezimalsystem korrekte Berechnungen erlaubt. Dadurch wurde der Code zur Berechnung der Nachfrage etwas weniger lesbar, da für diesen Datentyp anstatt von mathematischen Operatoren Methoden implementiert sind, welche diese Aufgabe übernehmen. Dieser Fehler trat nur an dieser Stelle auf, da es bei der Nachfrage mit Dezimalbrüchen gearbeitet wird, die eben diesen Fehler verursachten. Anderenfalls wäre schon die Deklaration der Variablen **demand** als **BigDecimal** eine elegantere Lösung gewesen, anstatt lokal eine Typkonvertierung vorzunehmen. Zur besseren Lesbarkeit wird die Berechnung der Nachfrage in Formel 10 explizit mit mathematischen Operatoren dargestellt.

```

public void insertDemandIntoNodes
(ProductionPlanModelEntity plan, Map<String, Node> tree,
Double[] ContingentPerPeriod, Double[] DeviationOfDemandPerPeriod,
int period, String NodeID) {
    for (int childCounter = 0; childCounter < plan.getCardinality();
        childCounter++) {
        Node n = tree.get(NodeID); // getting the node
        int np = n.getPeriod(); // Period
        n.setExpectedDemand(ContingentPerPeriod[np]);
        // BigDecimal to prevent ugly recurring decimals because of
        // internal binary representation
        BigDecimal demand = BigDecimal.valueOf(ContingentPerPeriod[np]).
            multiply(((BigDecimal.valueOf(plan.getSpreadOfDemandPerPeriod()
            [np]).divide(BigDecimal.valueOf(100)).add(BigDecimal.ONE)).
            subtract(BigDecimal.valueOf(DeviationOfDemandPerPeriod[np])).
            multiply(BigDecimal.valueOf(n.getChildCounter()))));

        n.setDemand(demand.doubleValue());
        tree.put(n.getId(), n); // updating the node

        String childNodeID =
            NodeID.concat(String.valueOf(childCounter));

        if (np < plan.getPeriods() - 1) { // if max. period
            insertDemandIntoNodes(plan, tree, ContingentPerPeriod,
                DeviationOfDemandPerPeriod,
                period + 1, childNodeID);
        }
    }
}

```

**Codefragment 4: Nachfrage in Node einfügen**

An dieser Stelle erkennt man auch den additiven Random Walk aus Abschnitt 2.5.2 und Formel 8 wieder. **ContingentPerPeriod** stellt dabei den Ausgangsparameter  $a_t$ , den Erwartungswert für die Berechnung der Nachfrage pro Periode, dar. Aus **getSpreadOfDemandPerPeriod()[np]** in Verbindung mit **DeviationOfDemandPerPeriod[np]** und **childCounter** wird  $\tilde{z}_j$  als Abweichung vom Erwartungswert berechnet.

$$\begin{aligned}
 demand &= ContingentPerPeriod[np] \\
 &+ plan.getSpreadOfDemandPerPeriod()[np] \\
 &- DeviationOfDemandPerPeriod[np] \\
 &* childcounter
 \end{aligned}$$

**Formel 10: Berechnung der Nachfrage**

Der Algorithmus aus Codefragment 4 geht dabei folgendermaßen vor. Nachdem für jede Periode ein gewisses Kontingent als Erwartungswert der Nachfrage zu produzieren ist und die meisten Knoten von diesem Erwartungswert durch einen Variationskoeffizient abweichen, wird diese Abweichung vom Erwartungswert hinzu addiert. Aus der erwarteten Nachfrage sowie dem Variationskoeffizienten

lässt sich eine Bandbreite berechnen, innerhalb derer die Nachfragewerten der einzelnen Geschwisterknoten mit demselben Elternknoten streuen. Nachdem die Anzahl der Knoten ja bekannt ist, lässt sich innerhalb dieser Bandbreite eine Schrittlänge errechnen, welche beginnend mit dem ersten Geschwisterknoten, der die maximal mögliche Nachfrage speichert, für jeden weiteren Knoten diese Schrittlänge mit `childCounter` multipliziert und von der maximalen Nachfrage subtrahiert. Dadurch wird die Streuung gleichmäßig auf alle Geschwisterknoten verteilt.

Anhand nun errechneten Nachfragewerte für jeden Knoten kann nun die optimale Produktionsmenge sowie auch die Lagerstände in jedem einzelnen Knoten ermittelt werden. Dabei spielt die Umsatzergebnisfunktion (Formel 5) aus Abschnitt 2.4.1 dieser Arbeit die zentrale Rolle. Die optimale Produktionsmenge wird folgendermaßen ermittelt. Der Anwender kann bei der Erstellung des Planes die Produktionsmenge innerhalb eines Minimums und Maximums beschränken sowie eine Schrittweite für die Änderung der Produktionsmenge innerhalb dieser Schranken angeben. Diese Angaben finden sich in `ProductionPlanModelEntity` wieder und werden dazu benutzt ein Array (`possibleProduction[]`) zu befüllen, welche alle möglichen Produktionsmengen enthält. Dazu werden zwei Methoden benötigt. Eine berechnet das Umsatzergebnis und die Zweite iteriert über die möglichen Produktionsmengen und sucht dabei jene mit dem optimalen Umsatzergebnis.

Die Methode, welche das Umsatzergebnis berechnet, wird in Codefragment 5 abgebildet. An dieser Stelle sei auf die Ermittlung des Umsatzergebnisses (Formel 5) aus Abschnitt 2.5.1. als Grundlage für die Berechnung hingewiesen.

Nachdem im SOPPS-Modell, das durch technologische Grenzen bei der Wahl der Produktionsmengen beschränkt wird, die abgesetzte Menge nicht der produzierten Menge entsprechen muss, erfordert eine Berechnung von Lagermenge und Umsatzergebnis gewisser Einschränkungen. Dazu dienen eine Reihe von if-else-Anweisungen, um den Absatz (`overturn`) einerseits und davon abhängig das Lager (`storage`) andererseits auf plausible Werte zu beschränken. Der Absatz kann dabei entweder der Produktionsmenge zusammen mit dem aus der Vorperiode stammenden Lagerstand entsprechen, oder der Nachfrage.

```

public Double calculateProfitOfNode(Double demand, Double
expectedDemand, Double supply, Double pastStorage, Double price, Double
fixedCosts, Double variableProductionCosts, Double
variableStorageCosts) {

    Double profit;
    Double overturn = 0.0;
    Double storage = 0.0;
    Double contributionMargin = price - variableProductionCosts;

    Double spS= supply + pastStorage;
    if (demand < spS) {
        overturn = demand;
    }
    else{
        overturn = spS;
    }

    Double spSd = supply + pastStorage - demand;
    if (spSd > 0.0) {
        storage = spSd;
    }
    else {
        storage = 0.0;
    }

    profit = (overturn * contributionMargin) -
            (variableStorageCosts * storage) - fixedCosts;
    return profit;
}

```

#### Codefragment 5: Ermittlung des Umsatzergebnisses

```

protected void calculateOptimalProductionPerPeriod
(ProductionPlanModelEntity plan, Map<String,Node> tree,
Double[] possibleProduction, String startNodeId)
{
    [...]//some variables for iterating over all nodes

    do
    {
        [...]//some variables out of node

        Double myOutput = null;
        Double myProfit = null;
        Double optimalOutput = null;
        Double optimalProfit = null;

        int aLength = possibleProduction.length;
        int i=0;

        do
        {
            myOutput = possibleProduction[i];

            myProfit = calculateProfitOfNode(demand, expectedDemand,
myOutput, pastStorage, price, fixedCosts, varProdCosts,
varStorCosts);
            if(myProfit > optimalProfit)
            {
                optimalProfit = myProfit;
                optimalOutput = myOutput;
                currentStorage = optimalOutput + pastStorage - demand;
            }
        }
        while (i < aLength);
    }
}

```

#### Codefragment 6: Bestimmung der optimalen Produktionsmenge

Codefragment 6 zeigt die Methode, welche die optimale Produktionsmenge aus `possibleProduction[]` auswählt. In einer Schleife wird jeder Wert des Arrays durchlaufen, das Umsatzergebnis berechnet und bei jeder Iteration gespeichert, sofern es größer dem aus der letzten Iteration ist. So werden das optimale Umsatzergebnis und damit die optimale Produktionsmenge ermittelt. Nachdem die Lagermenge direkt von der Produktionsmenge abhängig ist, muss dieser Wert ebenfalls schon an dieser Stelle berechnet und gespeichert werden.

Zur Ermittlung eines intertemporalen Umsatzerfolges sind Erwartungswerte des selbigen für jede Trajektorie innerhalb des Wahrscheinlichkeitsraumes notwendig. Die letzte Periode des Produktionsplanes stellt dabei einen Sonderfall dar, da dies ja keine Erwartungswerte von zukünftigen Perioden mehr berücksichtigen kann, da der Planungshorizont an dieser Stelle erreicht ist. Anstatt dessen werden die Umsatzergebnisse aller Geschwister aufsummiert und gemittelt. Genau diese Funktionalität erfüllt die Methode in Codefragment 7.

```

public void summarizeProfitofSiblingNodes (Map<String, Node> tree,
                                         String startNodeId, int periods, int cardinality)
{
    double totalProfit = 0.0; // total profit of descendant nodes
    for (int childCounter = 0; childCounter < cardinality;
         childCounter++) {
        String siblingNodeId = startNodeId.concat (String
            .valueOf(childCounter));

        // to handle TotalExpectedDemand for last period of the plan
        if (tree.get(siblingNodeId).getPeriod() == periods - 1) {
            Double tep = 0.0;
            for (int i = 0; i < cardinality; i++) {
                String currentNodeId = siblingNodeId.substring(0,
                    siblingNodeId.length() - 1).concat (String.valueOf(i));
                tep += tree.get(currentNodeId).getProfit();
            }
            tep /= cardinality;
            for (int i = 0; i < cardinality; i++) {
                String currentNodeId = siblingNodeId.substring(0,
                    siblingNodeId.length() - 1).concat (String.valueOf(i));
                Node n = tree.get(currentNodeId);
                n.setTotalExpectedProfit(tep);
                tree.put(currentNodeId, n);
            }
        }

        else if (tree.get(siblingNodeId).getPeriod() < periods - 1) {
            summarizeProfitofSiblingNodes(tree, siblingNodeId, periods,
                cardinality);
        }
    }
}

```

**Codefragment 7: Ermittlung des erwarteten Umsatzergebnisses**

Erst nachdem all diese Werte berechnet wurden, kann schlussendlich die Kernfunktionalität der SOPPS ausgeführt werden, die intertemporale Optimierung des Umsatzergebnisses. Diese wird wie nach Papageorgiou in Abschnitt 2.5.3.



dieser Arbeit vom Planungshorizont rückwärts fortschreitend bis zur Gegenwart berechnet.

Dies geschieht, indem die erwarteten Umsatzergebnisse der Geschwisterknoten der letzten Planungsperiode aus dem vorhergehenden Schritt gewichtet nach Wahrscheinlichkeiten summiert werden. Im derzeitigen Entwicklungsstand des Prototyps sind die einzelnen Zustände des Wahrscheinlichkeitsraumes allerdings noch gleichverteilt, weshalb dies dem Mittelwert der Umsatzergebnisse entspricht. Dies führt logischerweise zu keiner Änderung für die letzte Periode, allerdings beginnt nun die Bestimmung des intertemporalen Umsatzergebnisses über alle Planungsperioden hinweg. Der Erwartungswert der letzten Periode wird mit dem Umsatzergebnis des Elternknotens der vorhergehenden Periode addiert. Damit erhält man einen Wert, welcher das erwartete Umsatzergebnis für zwei Perioden angibt, da ja einige Arbeitsschritte zuvor die Produktionsmenge anhand der erwarteten Nachfrage bestimmt wurde.

Diese Prozedur wird rückwärts fortschreitend bis in die Gegenwart für alle Knoten wiederholt. Der Erwartungswert des Umsatzergebnisses in jedem Knoten zeigt nun den Erwartungswert für die verbleibenden zukünftigen Planperioden.

```
public void summarizeTotalProfitOfSiblingNodes (Map<String, Node> tree,
                                              String startNodeId, int periods, int cardinality)
{
    double totalProfit = 0.0; // total profit of descendant nodes
    double currentProfit = tree.get(startNodeId).getProfit();
    for (int childCounter = 0; childCounter < cardinality;
         childCounter++) {
        String siblingNodeId = startNodeId.concat
            (String.valueOf(childCounter));

        if (tree.get(siblingNodeId).getPeriod() < periods - 1) {
            summarizeTotalProfitOfSiblingNodes(tree, siblingNodeId,
                                              periods, cardinality);
        }
        totalProfit += tree.get(siblingNodeId).getTotalExpectedProfit();
    }
    ...}
    totalProfit /= cardinality;
    totalProfit += currentProfit;
    Node n = tree.get(startNodeId);
    n.setTotalExpectedProfit(totalProfit);
    tree.put(startNodeId, n);
}
```

#### Codefragment 8: Berechnung des intertemporalen Umsatzergebnisses

Bei der Filtrierung des Wahrscheinlichkeitsraumes werden nun die Vorteile dieser Lösung deutlich. Bei der im Laufe des Zeitablaufes stattfindenden Informationsenthüllung kann auf diese Art und Weise die zukünftige Entwicklung des Umsatzergebnisses immer genauer abgeschätzt werden, da Werte aus nicht

mehr erreichbaren und damit für die Produktionssteuerung irrelevanten Trajektorien im Wahrscheinlichkeitsraum nicht mehr berücksichtigt werden.

### **3.3.3. Prozessorientierte Benutzeroberfläche**

Wie bereits bei der Vorstellung von ERP-Control sowie bei der verwendeten Technologien erwähnt, spielt Prozessorientierung eine essenzielle Rolle bei der Implementierung der Applikation. Dank jBPM lässt sich ein GP graphisch, mitunter auch von der betreffenden Fachabteilung selbst, modellieren. Dieses Modell kann dann direkt weiter verwendet werden um die GP zu implementieren. Dieser Grundgedanke zieht sich durch die gesamte Applikation und wurde so auch im Programmmodul zu dieser Arbeit umgesetzt.

Nachfolgend werden die einzelnen Funktionen des SOPPS-Moduls vorgestellt. Anhand der Prozessmodelle wird der Anwender durch einen GP geführt. Dadurch kann sichergestellt werden, dass jeder Schritt in der richtigen Reihenfolge ausgeführt wird und nicht übersehen wird, bevor ein GP abgeschlossen werden kann. Wo es möglich ist, werden die Eingaben des Anwenders auch auf Plausibilität überprüft, um mögliche Fehleingaben von vorn herein auszuschließen.

Ganz eliminieren lässt sich die Fehlerquelle Mensch natürlich nicht, aber das soll auch nicht der Zweck der Applikation sein sondern vielmehr soll sie dem Anwender ein Werkzeug in die Hand geben, um die Arbeit zu erleichtern und in strukturierte Bahnen zu lenken.

Dieser Abschnitt gliedert sich folgendermaßen. Der Ablauf jeder Funktion des SOPPS-Moduls wird zuerst anhand eines jBPM-Diagramms erläutert. Dessen Umsetzung in der Benutzeroberfläche wird dann im Anschluss durch Screenshots der Applikation erläutert.

### 3.3.3.1. Generierung und Modifikation des Produktionsplans

Bevor das SOPPS-Modul überhaupt seinen vollen Funktionsumfang ausspielen kann, muss man selbstredend zu aller erst Produktionspläne erstellen. Die Erstellung eines solchen Planes ist in der Abbildung 17 mit jBPM modelliert worden. Die einzelnen Aufgaben innerhalb des Prozesses werden durch Task Nodes dargestellt

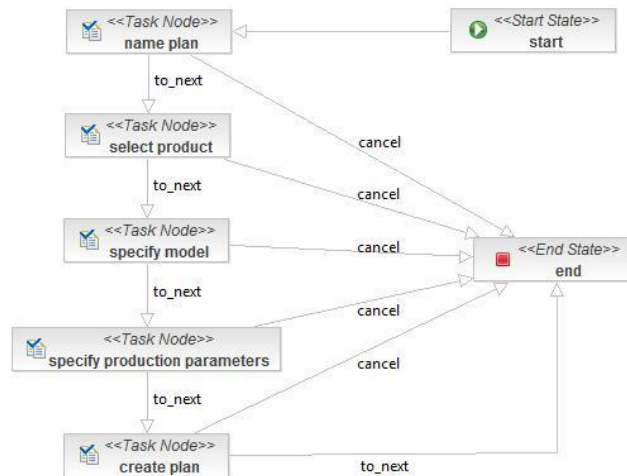


Abbildung 17: Geschäftsprozess der Plangenerierung

Nachdem eine neue Prozessinstanz aufgerufen wurde, muss man den Plan im ersten Schritt benennen. Hierbei wird bereits überprüft, ob es bereits einen Plan mit demselben Namen in der Datenbank gibt. Ist dies der Fall wird der Anwender durch eine Fehlermeldung darauf hingewiesen. Im nächsten Schritt wird der Anwender aufgefordert, das zu produzierende Produkt aus einer Liste auszuwählen. Dann erfolgt die Spezifikation der Modellparameter, also wie der Baum aussieht. Nach Eingabe der Produktionsparameter wird der Plan in Form des Baumes generiert. Der Anwender kann nochmals seine Eingaben überprüfen und den Baum speichern.

In jedem Schritt hat der Anwender die Möglichkeit, die Prozessinstanz abubrechen. In diesem Falle wird die entsprechende Session Bean mitsamt aller bisherigen Eingaben gelöscht. Die Session Bean speichert die Zustände. Wird der Prozess also nicht bis zum Ende durchgeführt und auch nicht abgebrochen, bleiben die Eingaben erhalten und der Anwender kann den Prozess zu einem späteren Zeitpunkt vollenden.

Wie in der GUI die Erstellung eines Produktionsplanes dargestellt wird, zeigt die nächste

Abbildung 18. Nach der Initialisierung einer neuen Instanz des GP werden wie in der Abbildung ersichtlich die Optionen für die aktuelle Aufgabe des Prozesses angezeigt. Hat der Anwender eine Aufgabe erledigt, werden die Optionen für die nächste Aufgabe sichtbar. Auf diese Weise wird der Anwender Schritt für Schritt durch den GP geführt.

The screenshot shows a window titled "Strategic Production Planning Creation". At the top, there is a link "Initialize new Business Process Instance". Below this, the window is divided into five horizontal sections, each representing a step in the process:

- Name plan**: Contains a text input field and two buttons: "enter name of the plan" and "cancel".
- Select product**: A section for selecting a product.
- Specify model parameters**: A section for specifying model parameters.
- Specify production parameters**: A section for specifying production parameters.
- Create Plan**: A section for creating the plan.

Abbildung 18: Ablauf Plangenerierung

Im Falle mehrerer offener Instanzen eines GP werden in den einzelnen Aufgaben je eine Zeile von Optionen für je eine Instanz in der Reihenfolge der Eröffnung angezeigt. Wählt der Anwender eine Aufgabe aus, so wird der Ablauf des GP ausgeblendet und nur die zur Erfüllung der Aufgabe notwendigen Eingabefelder oder Daten werden angezeigt. Folgende Abbildung zeigt dabei die Benennung eines neuen Produktionsplanes. Gibt der Anwender einen Namen ein, der bereits in der Datenbank gespeichert ist, wird ihm das durch eine Fehlermeldung angezeigt. Mehrere gleichnamige Pläne würden andernfalls nur unnötig Verwirrung stiften.

The screenshot shows a window titled "Name plan". It contains a label "Plan Name:" followed by a red asterisk, indicating a required field. To the right of the label is a text input field. Below the input field is a blue button labeled "save".

Abbildung 19: Plan benennen

Im nächsten Schritt muss der Anwender ein Produkt auswählen, das produziert werden soll. Dafür werden neben der ID des Produkts und dessen Beschreibung auch der aktuelle Lagerstand sowie die Maßeinheit angezeigt. Das soll verhindern, dass der Anwender einen Plan für ein Produkt erstellt, für das ohnehin genügend Lagerbestand vorrätig ist.

Product Selection				
Product ID	Description	Actual Stock	UoM	
1	Guss-Paraffin	500000.00	kg	<a href="#">Select</a>
2	Press-Paraffin	500000.00	kg	<a href="#">Select</a>
3	Zug-Paraffin	500000.00	kg	<a href="#">Select</a>
4	Tauch-Paraffin	500000.00	kg	<a href="#">Select</a>
5	Farbe	200.00	kg	<a href="#">Select</a>
6	Docht Guss-Kerzen	100.00	kg	<a href="#">Select</a>
7	Docht Press-Kerzen	100.00	kg	<a href="#">Select</a>
8	Docht Zug-Kerzen	100.00	kg	<a href="#">Select</a>
9	Gusskerzen	88896.00	kg	<a href="#">Select</a>
10	Presskerzen	14763.00	kg	<a href="#">Select</a>
11	Zugkerze	67898.00	kg	<a href="#">Select</a>

Abbildung 20: Produktauswahl

Abbildung 21 zeigt die Modellparameter, die der Anwender spezifizieren muss. Hier wird angegeben, über wie viele Perioden (Periods) der Produktionsplan laufen soll sowie die Anzahl der Kinderknoten für das Baummodell. Die Anzahl (Cardinality) wurde auf minimal zwei und maximal zehn Kinder beschränkt, da einerseits eine Planung mit nur jeweils einem möglichen Zustand in der zukünftigen Periode den stochastischen Hintergrundgedanken von SOPPS ad absurdum führen würde und andererseits die Programmlogik nur auf bis zu zehn Kinderknoten ausgelegt ist. Die Gründe dafür wurden bereits ausführlich in Kapitel 3, Unterpunkt 1, diskutiert. Versucht der Anwender dennoch ein Modell mit mehr als 10 Kinderknoten zu erstellen, so wird er durch eine Fehlermeldung darauf hingewiesen und die Vollendung der Aufgabe wird solange blockiert, bis der Anwender eine gültige Cardinality für das Modell eingegeben hat.

**Specify model parameters**

Periods: \*

Cardinality: \*

Abbildung 21: Modellparameter spezifizieren

Nach der Eingabe der Eingabe gültiger Modellparameter wird der Anwender zur Eingabemaske für die Produktionsparameter geführt. Diese wird in der nachfolgenden Abbildung 22 illustriert. Bei jedem der folgenden Eingabefelder wird überprüft, ob der Anwender Zahlen eingegeben hat. Alles andere wäre sinnlos und wird durch eine Fehlermeldung angezeigt.

Die erste Reihe von Feldern (Percentage of demand per period) repräsentiert die anteilmäßige Verteilung des Gesamtbedarfes auf die vier Quartale eines Jahres in Prozent. Diese Felder sind bereits standardmäßig mit Erwartungswerten errechnet aus historischen Daten in der Datenbank befüllt. Der Anwender kann diese jedoch korrigieren, falls er für den Planungszeitraum eine andere Verteilung der Nachfrage übers Jahr erwartet.

Darunter kann der Anwender angeben, in wie weit der Plan um die Erwartungswerte der Nachfrage (Spread of Demand per period) in Prozent streuen soll, also in wie weit der Planungsalgorithmus eine Abweichung vom Erwartungswert nach dem additiven RW aus Abschnitt 2.5.2 berücksichtigen soll. Auch diese Eingabefelder sind bereits mit historisch berechneten Werten befüllt. Bei den vier Perioden handelt es sich auch diesmal um die Quartale eines Jahres.

Specify production parameters				
Percentage of demand per period	20.0	- 14.0	- 25.0	- 41.0
Spread of demand per period:	12.0	- 15.0	- 10.0	- 5.0
Period and year of start date: *	1			
	00			
Product price:	26.87			
Variable production costs: *	10.0			
Fixed Costs: *	100000.0			
Storage costs: *	3.0			
Planned production quantity: *	40000.0			
Minimum production: *	500.0			
Maximum production: *	21000.0			
Production step: *	1000.0			
Start storage: *	100.0			
<input type="button" value="save"/>				

Abbildung 22: Produktionsparameter spezifizieren

Laut ECSI-Standard muss ein Produktionsplan einen klar definierten Beginn und Ende haben. Das Ende ergibt sich implizit aus der Angabe der Perioden in den Modellparametern. Der Beginn eines Planes wird eingegeben durch die Auswahl des entsprechenden Quartals und des Jahres. Die Auswahl der Quartale ist auf eins bis vier beschränkt, um Fehleingaben auszuschließen.

Weiters folgen diverse Angaben für die Produktion. Der Produktpreis (Product price) ist wiederum bereits durch den aktuellen Preis des Produktes in der Datenbank ausgefüllt. Der Anwender kann diesen natürlich ändern. Die variablen Produktionskosten (Variable production costs) spiegeln die Produktionskosten für eine Einheit des zu produzierenden Produktes wider. Die Fixkosten (Fixed costs) stellen die nicht variablen Produktionskosten dar, die in jedem Fall entstehen, auch wenn nichts produziert wird. Die variablen Lagerkosten (Storage costs) geben wiederum die Lagerkosten für eine Einheit des Produktes an. Diese Werte sind erforderlich für die spätere Erfolgsbewertung der einzelnen Knoten des Baumes.

Es folgen mengenmäßige Angaben zur Produktion. Zu aller erst muss der Anwender die geplante Gesamtproduktionsmenge (Planned production quantity) angeben. Diese wird in nachfolgenden Berechnungen durch Percentage of Demand per Period anteilmäßig auf die einzelnen Perioden verteilt und durch Spread of Demand per Period wird die Abweichung davon auf die einzelnen Knoten des Planes aufgeteilt.

Um technologischen Einschränkungen Rechnung zu tragen ist der Anwender auch in der Lage mögliche Produktionsmengen anzugeben. Die minimale Produktion (Minimum production) gibt dabei die minimale Produktionsmenge an. Maximum production repräsentiert die maximale Produktionsmenge. Da mitunter die Produktionsmenge nur diskret angepasst werden kann, muss der Anwender ein Intervall (Production Step) angeben, um das die Produktion gesteigert bzw. verringert werden kann. Abschließend muss die aktuelle Lagermenge (Start Storage) angegeben werden, die ja ebenfalls für die Produktionsmenge der folgenden Intervalle relevant ist. Auch dieses Feld ist bereits mit dem aktuellen Wert aus der Datenbank für das jeweilige Produkt befüllt, kann jedoch geändert werden, sofern der Anwender eine andere Lagermenge zum Startzeitpunkt des Planes erwartet.

Bei der Eingabe der Verteilung der Nachfrage über die Quartale findet eine Plausibilitätsüberprüfung statt. Die Summe der Prozentangaben darf dabei wie in der folgenden

Abbildung 23 ersichtlich nicht 100 % überschreiten.

Specify production parameters

Percentage of demand per period: 33 - 33 - 33 - 33 ❌ The sum of the percentages must be 100.

Spread of demand per period: 12.0 - 15.0 - 10.0 - 5.0

Abbildung 23: Fehlermeldung bei nicht plausiblen Werten

Dabei spielt es keine Rolle, ob bei den Modellparametern der Planungszeitraum länger oder kürzer als ein Jahr angegeben wurde. Die Verteilung bezieht sich immer auf ein ganzes Jahr und dieses hat vier Quartale. Das ist auch sinnvoll, wenn ein Plan z.B. erst im dritten Quartal eines Jahres beginnt. In dem Falle muss sich der Anwender keine Gedanken machen, ob das dritte Quartal eines Jahres die erste Planungsperiode darstellt. Die Werte beziehen sich stets auf ein Jahr. Auf diese Weise werden auch Fehleingaben vermieden, da der Anwender die Verteilung immer auf dieselbe Weise eingibt, egal wann der Plan startet, wie lange er dauert oder wann er endet.

Im nächsten Schritt dieses GP kann der Anwender seine Eingaben in einer Übersicht ein letztes Mal überprüfen.

Create Plan

save plan cancel

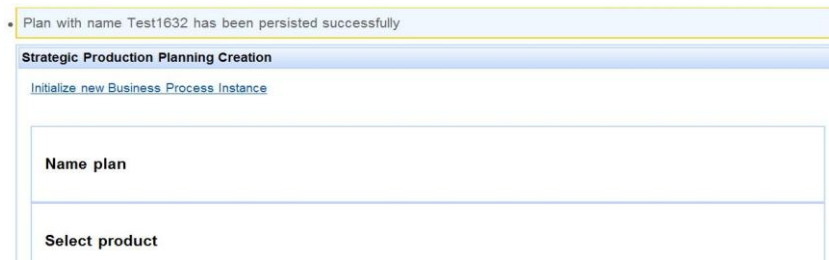
Input parameters

Plan name	Test1632
Cardinality	2
Periods	4
Percentage of demand per period	20.0 - 14.0 - 25.0 - 41.0

Abbildung 24: Plan speichern

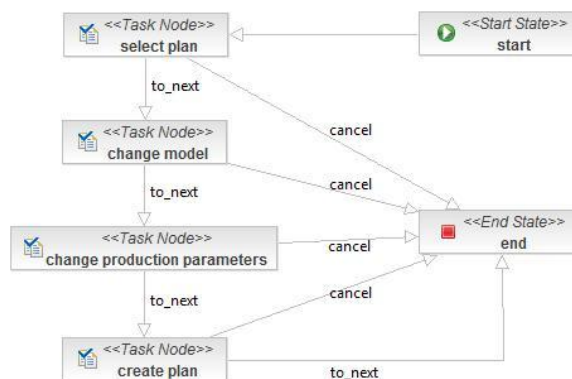
Dann muss er den Plan nur mehr speichern, oder kann den Prozess abbrechen. Im Falle einer Speicherung wird abschließend eine Erfolgsmeldung ausgegeben, wie in folgender Abbildung ersichtlich ist.





**Abbildung 25: Bestätigung des gespeicherten Plans**

Der GP zur Änderung eines bereits gespeicherten Planes wird in der Abbildung 26 dargestellt. Die meisten Aufgaben sind bereits aus der Planerstellung bekannt und werden deshalb auch nicht erneut besprochen.



**Abbildung 26: Geschäftsprozess das Planmodifikation**

Es gibt dennoch zwei Unterschiede zur Erstellung eines Produktionsplanes. Anstatt als erste Aufgabe des Prozesses einen Plan zu benennen, wird ein bereits gespeicherter Plan aus einer Liste ausgewählt. Dabei werden jedoch nur jene Pläne aufgelistet, welche noch nicht genehmigt sind (siehe nächster Prozess). Auch ein Produkt muss nicht mehr ausgewählt werden, da davon ausgegangen werden kann, dass sich das Produkt, für welches die Produktion geplant wird, nicht mehr ändern wird, sondern nur die Parameter des Modells oder der Produktion.

### 3.3.3.2. Genehmigung des Plans

Damit nicht jeder generierte Plan in der Produktion umgesetzt wird, müssen, bevor die geschieht, entsprechende Produktionspläne vorab erst genehmigt werden. Dabei wird im Prinzip nur das approved-Attribut in der entsprechenden Instanz von ProductionPlanModelEntity auf true gesetzt und in der Datenbank gespeichert. Die Genehmigung an sich stellt allerdings auch einen eigenständigen GP dar, welcher eben als solcher modelliert und implementiert wurde. Abbildung 27 zeigt dabei den Ablauf, der im Wesentlichen nur aus der Auswahl eines Planes und dessen Genehmigung besteht.

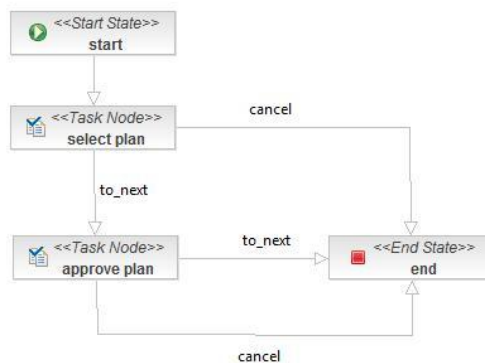


Abbildung 27: Geschäftsprozess Genehmigung des Plans

Auch in diesem Fall wird vom Anwender wieder ein GP angestoßen, welcher sich dem Anwender Schritt für Schritt in  
Abbildung 28 darstellt. In  
Abbildung 29 wird wie erwähnt ein Plan ausgewählt.



Abbildung 28: Ablauf Plangenehmigung

Select plan		
Plan name	Start date	
testme	2025-02-01 20:47:57.0	<a href="#">select</a>
Test1632	2000-02-01 17:12:15.0	<a href="#">select</a>
test1754	2003-02-01 17:57:13.0	<a href="#">select</a>
test1825	2004-08-01 18:12:15.0	<a href="#">select</a>

**Abbildung 29: Auswahl des Planes**

Im letzten Schritt des Prozesses werden wie in Abbildung 30 ersichtlich die Modell- und Produktionsparameter des Plans zur finalen Kontrolle durch den Anwender zusammenfassend angezeigt bevor dieser den Plan durch klicken auf den approve-Button genehmigt. Abschließend erfolgt analog zur Erstellung eines Plans die Ausgabe einer Erfolgsmeldung.

Approve plan	
Plan name	Test1632
Cardinality	2
Periods	4
Percentage of demand per period	20.0 - 14.0 - 25.0 - 41.0
Spread of demand per period	12.0 - 15.0 - 10.0 - 5.0
Period and year of start date	2000-02-01 17:12:15.0
Prodcut price	26.87
Variable production costs	10.0
Fixed Costs	100000.0
Variable storage costs	3.0
Planned production quantity	40000.0
Minimum production	500.0
Maximum production	21000.0
Production step	1000.0
<input type="button" value="approve"/> <input type="button" value="cancel"/>	

**Abbildung 30: Übersicht und Bestätigung des Plans**

### 3.3.3.3. Simulation und Durchführung des Plans

Die Simulation und auch die Durchführung eines Planes funktionieren an sich nach demselben Schema. Der einzige Unterschied besteht darin, dass in der Simulation nur nicht genehmigte Pläne ausgewählt und simuliert werden können und in der Durchführung analog nur genehmigte Pläne auch ausgeführt werden können. Der Ablauf dieses GP wird in den Abbildungen Abbildung 31 und Abbildung 32 gezeigt.

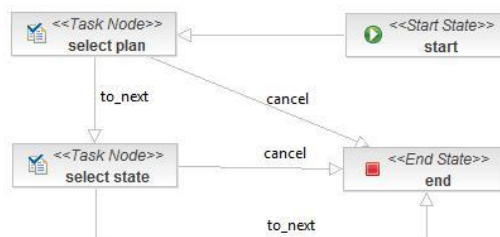


Abbildung 31: Geschäftsprozess Simulation des Plans

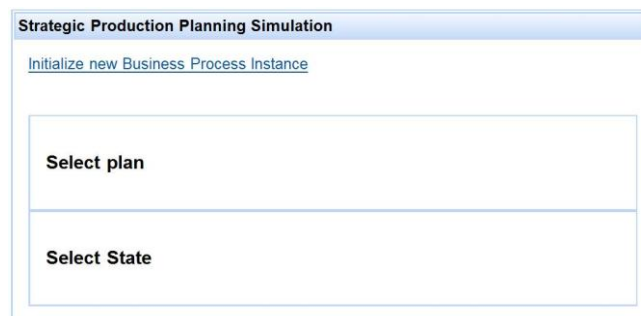


Abbildung 32: Ablauf der Simulation

Um dem Anwender zusätzlich zum Baum mehr Übersicht über die wichtigsten Daten des Planes zu verschaffen, wird zusätzlich eine Tabelle angezeigt. In Abbildung 33 sieht man, wie viel auf Grund welcher erwarteten Nachfrage produziert wurde und welches Umsatzergebnis dabei erzielt wurde.

Select state				
finish				
Period	Expected demand	Production output	Compliance	Profit
1	23000,00	1000,00	1,15	250260,00
2	15000,00	7000,00	1,00	15000,00
			Total profit	265260,00

Abbildung 33: Übersichtstabelle des Planfortschritts

Der Baum stellt sich bei Simulation und Durchführung auszugsweise wie in Abbildung 34 dar. Hier spiegelt sich auch die Filtration des Wahrscheinlichkeitsraumes wieder. Knoten, welche nicht mehr erreichbar sind, werden grau dargestellt und sind damit für den Anwender klar erkennbar. In der kommenden Periode im Planverlauf findet sich in den Knoten ein Select-Button wieder. Wird dieser vom Anwender angeklickt findet erneut eine Filtration statt und die Anzahl der möglichen Knoten reduziert sich weiter bis in der letzten Periode nur mehr ein Knoten übrig geblieben ist. Die bis dahin realisierte Trajektorie im Wahrscheinlichkeitsraum wird gekennzeichnet durch realisierte Knoten, welche weiterhin in schwarz dargestellt werden, allerdings ohne den select-Button.

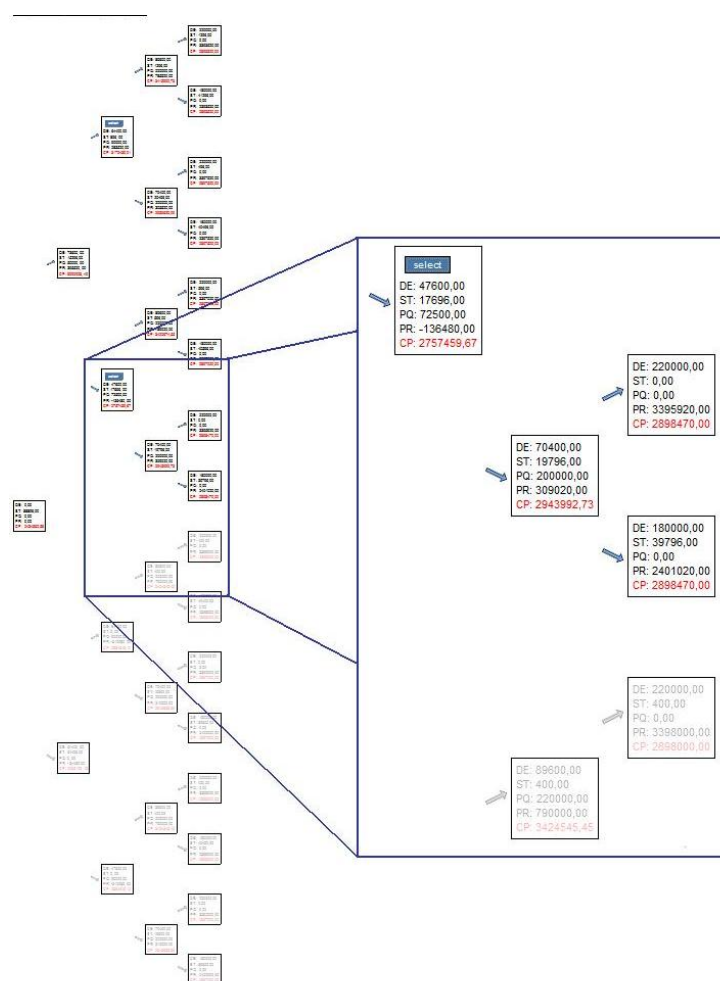


Abbildung 34: Darstellung und Auswahl der Baumknoten

Die Werte in den einzelnen Knoten stehen dabei in der Reihenfolge von oben nach unten für die Nachfrage in dem betrachteten Zustand, dessen Lagerstand, die ermittelte optimale Produktionsmenge, dem Umsatzergebnis in dem Zustand sowie den kumulierten Barwert des Betriebsergebnisses der von diesem Zustand aus erreichbaren Trajektorien in Abhängigkeit vom Zustand selbst. Letzteres ist von besonders zentraler Bedeutung für den Anwender und stellt auch das Kernstück dieser Arbeit dar. Deshalb wird dieser Wert auch mit roter Schriftfarbe besonders hervorgehoben.

Die stochastische Kontrolltheorie soll es ja ermöglichen eine intertemporal optimale Planung zu verwirklichen. Die Betrachtung der einzelnen Ergebnisse in jedem Zustand für sich führen nicht unbedingt zu optimalen Entscheidungen über den gesamten Planungszeitraum hinweg. Die Betrachtung von Einzelergebnissen führt viel mehr zu einer lokalen Optimierungsstrategie, wobei das Gesamtziel mitunter aus den Augen verloren wird. Durch eine längerfristige Betrachtung über mehrere Perioden hinweg können bessere Umsatzergebnisse erzielt werden, da die Entscheidungsträger lokal suboptimale Ergebnisse in Kauf nehmen, um dafür über den gesamten Planungszeitraum hinweg ein besseres Ergebnis erzielen zu können. Diese Betrachtungsweise erlaubt es, wie schon in der Einleitung erwähnt, Chancen zu entdecken und auch zu nutzen, die andernfalls brach liegen würden. Aus diesem Grund ist ein Wert, welcher zur Bewertung des Erfolges einer Unternehmung über mehrere Perioden hinweg dient, von so zentraler Bedeutung.

## **4. Resümee**

Das letzte Kapitel dieser Arbeit erläutert zum Abschluss die Ergebnisse der selbigen noch einmal kurz und geht dabei auf den Leistungsumfang ein, den die modellbasierte Planung und Steuerung innerhalb von ERP-Control leistet. Es werden Vor- und Nachteile der Lösung kurz beleuchtet, welche teilweise auch auf Vereinfachungen im Modell zurückzuführen sind. Daraus ergeben sich auch diverse unbeantwortete Fragen und etwaige Probleme, deren Lösung offen bleibt. Die Zusammenfassung gibt noch einmal einen kurzen Überblick über die gewonnenen Erkenntnisse aus der Recherche und welche Zusammenhänge sich in den eingangs vorgestellten Konzepten heraus kristallisierten. Beendet wird dieses Kapitel mit einem vorsichtigen Blick in die Zukunft. Hier sollen kurz die Chancen und Möglichkeiten beschrieben werden, die sich durch eine Weiterentwicklung von ERP-Control ergeben könnten und welche Verbesserungen und Erweiterungen eventuell sinnvoll wären.

### **4.1. Analyse des Lösungsansatzes**

Der folgende Abschnitt soll den erarbeiteten Lösungsansatz einer SOPPS zusammenfassen und evaluieren. Dazu wird erstens der Leistungsumfang noch einmal kurz dargelegt sowie die Stärken und Schwächen der Lösung diskutiert. Daraus ergeben sich dann offene Fragen, die bei einer Weiterentwicklung des vorgestellten Prototyps berücksichtigt werden sollten bzw. könnten.

#### **4.1.1. Leistungsumfang**

Die implementierte Lösung einer modellbasierten Planung und Steuerung unter Unsicherheit soll auf Basis stochastischer Modelle einen Produktionsplan über mehrere Perioden realisieren, welcher intertemporal optimierte Produktionsmengen errechnet. Durch Einbezugnahme von Konzepten aus der Kybernetik, insbesondere der kybernetischen Steuerung, wird so ein selbstregulierender Steuerkreis geschaffen. Die Berücksichtigung diverser in der Wirtschaft etablierter Standards und Normen soll dabei der Akzeptanz der ERP-Software ERP-Control beitragen sowie auch größtmögliche Kompatibilität zu bestehender Unternehmenssoftware, welche diese Standards und Normen

implementieren. Dadurch werden Schnittstellen geschaffen, die eine Integration in bestehende Softwarearchitekturen ermöglichen und damit den Weg bereiten für einen späteren produktiven Einsatz, sobald ERP-Control die Prototypenphase hinter sich gelassen hat.

Durch die Produktionsplanung wurde ERP-Control um eine wichtige Eigenschaft ergänzt, die ein ERP-System als Funktionalität implementieren sollte. Die zukunftsorientierte Planung ist gerade in der heutigen Zeit, wo sich die Wirtschaft im Umschwung befindet und der Markt stetig turbulenter und unberechenbarer wird ein nicht zu unterschätzender strategischer Vorteil.

Die SOPPS versetzt ihren Anwender in die Lage, zukünftige für die Produktion relevante Entwicklungen besser zu erkennen und abzuschätzen, was ganz im Sinne von COSO II ist. Dabei darf allerdings nicht vergessen werden, dass der Anwender bei der Erstellung des Modells dessen Detaillierungsgrad und Kalibrierung selbst bestimmt. Es wird also davon ausgegangen, dass zukünftige Anwender ein grundlegendes Verständnis für den Anwendungsbereich mitbringen, und das SOPPS-Modell nicht mit unrealistischen Werten kalibrieren. Denn selbst das beste Modell vermag es nicht, zukünftige Entwicklungen mit akzeptabler Zuverlässigkeit zu prognostizieren, wenn die Ausgangswerte der Berechnungen nichts mit der Realität zu tun haben.

Der SOPPS-Prototyp erfüllt dabei folgende Aufgaben:

- Erstellung einer Baumstruktur (Modell)
- Kalibrierung des Modells mit Ausgangswerten
- Befüllung des Modells anhand der kalibrierten Werte (Produktionsplan)
- Berechnung einer stochastisch optimalen Produktionsmenge
- Informationsenthüllung im Zeitverlauf durch Neuberechnung des Plans
- Intertemporale Optimierung des Umsatzergebnisses
- Produktionssteuerung und/oder -simulation

Bei der Erstellung der Baumstruktur muss sich der Anwender entscheiden, wie detailliert der Plan strukturiert sein soll und über wieviele Perioden er verläuft. Er definiert also die Modellstruktur auf der der Plan aufsetzt. Dadurch wird nicht nur spezifiziert, wie viele Perioden in die Zukunft geplant wird, sondern auch wie viele Trajektorien den Wahrscheinlichkeitsraum aufspannen und wie weit die



unterschiedlichen Zustände voneinander wertmäßig entfernt sind. Je höher die Anzahl der Knoten in dem Planungsbaum ist, desto detaillierter kann auch geplant werden.

Bei der Kalibrierung des Modells wird festgelegt, wie hoch die Produktionsmenge sein soll und wie sich diese über die vier Quartale eines Jahres verteilt. Desweiteren kann der Anwender hier auch angeben, wie breit gefächert die Nachfragewerte der einzelnen Knoten sein sollen, also wie groß das Intervall der möglichen Nachfragemengen ist. Neben der Eingabe jener Parameter, die den Erlös sowie die Kosten der Produktion bestimmen wird auch auf technologische Parameter Rücksicht genommen, indem minimale und maximale Produktionsmengen sowie die möglichen mengenseitigen Anpassungen in der Produktion angegeben werden können.

Nachdem der Produktionsplan mehrere mögliche Zustände berücksichtigt, wird für jeden dieser Zustände jene Produktionsmenge errechnet, welche das Betriebsergebnis maximiert. Dabei werden auch mögliche Lagerzustände aus vergangenen Perioden in die Berechnungen mit einbezogen, die sich auf die betrachtete Periode auswirken können.

Durch die Informationsenthüllung im Zeitablauf werden die Vorhersagen über den erwarteten Gewinn der verbleibenden Perioden immer genauer. Zu Planbeginn werden in der Berechnung sämtliche Trajektorien berücksichtigt. Nachdem die erste Periode abgeschlossen ist, werden jene Trajektorien, welche nicht mehr durchlaufen werden können, von der Berechnung ausgeschlossen. Je näher man dem Planungshorizont kommt, desto genauer kann das Betriebsergebnis auch bestimmt werden. Dabei erhält der Anwender in einer Übersichtstabelle Informationen über die bereits vergangenen Planungsperioden und behält so das Planungsziel nicht aus den Augen und kann gleichzeitig die Abweichungen von Plan- und Ist-Werten ablesen. Diese Informationen sollen dazu dienen, die Planung in Zukunft weiter zu optimieren, da so auf Erfahrungswerte zurückgegriffen werden kann.

Das Betriebsergebnis wird nicht stationär für jede Periode einzeln, sondern intertemporal über alle Perioden betrachtet. Durch diese Sichtweise erhält der Anwender bei der Auswahl eines Knotens nicht nur Informationen über den möglichen Gewinn der kommenden Periode, sondern die Entwicklung des Umsatzergebnisses über alle zukünftigen Planperioden wird nicht aus den Augen

verloren. Die Entscheidung, welcher Weg innerhalb des Planungsbaumes eingeschlagen werden soll, basiert also nicht auf kurzfristigen Erwartungen, sondern erfolgt auf Grundlage einer längerfristigen Betrachtung der möglichen Entwicklungen dargestellt durch Erwartungswerte.

Von dieser ganzheitlichen Betrachtung profitiert auch die Simulation und Ausführung der erstellten Produktionspläne. In der Simulation können Szenarien in der zukünftigen Entwicklung der Nachfrage und der notwendigen Anpassungen in der Produktion durchgespielt werden. Diese Simulationen sind dazu gedacht, in den Entscheidungsprozess mit ein zu fließen, bevor ein Plan genehmigt und auch tatsächlich ausgeführt wird.

#### **4.1.2. Vor- und Nachteile der Lösung**

Die vorgestellte Lösung einer modellbasierten Planung und Steuerung gibt dem Anwender ein Werkzeug in die Hand, das es ihm ermöglicht eine Produktionsplanung eine stochastisch optimale Produktionsplanung und Steuerung ohne wesentliche Vorkenntnisse zu erstellen und durchzuführen. Anhand von Simulation und Durchführung von Plänen ist der Anwender in der Lage zu erkennen, welche Auswirkungen kleine Veränderungen der Modell- und Produktionsparameter auf die Produktion und in weiterer Folge auf den Umsatzergebnis haben. Durch einen prozessorientierten Ansatz in der Gestaltung der Benutzeroberfläche realisiert durch jBPM werden menschliche Fehler bei der Bedienung der Applikation soweit möglich vermieden und der Anwender wird Schritt für Schritt durch einen Geschäftsprozess geführt. Durch jBPM können auch Anwender aus Fachabteilungen zukünftig an der Gestaltung von ERP-Control mitwirken und praxisnah ihre Prozesse modellieren, die im weiteren Verlauf der Entwicklung auch als Grundlage für eine Implementierung verwendet werden können.

Durch Anwendung der REA-Ontologie kann der Anwender bei der Erstellung von Produktionsplänen sich voll und ganz auf die nötigen Absatzmengen konzentrieren, um die Nachfrage zu befriedigen. Zusätzliche Details wie Kapazitätsplanung oder Ressourceneinsatzplanung werden in der strategischen Planung ausgeblendet. Möglich macht dies die REA-Ontologie, bei der Ressourcen, Ereignisse und Agenten schon in der Datenstruktur miteinander

verknüpft sind. Diese Überlegungen werden in der Detailplanung adressiert, die jedoch nicht Teil dieser Arbeit ist.

Durch die Anwendung eines stochastischen Prozesses als Grundlage des Planungsbaumes ist der Plan keine Abfolge von Handlungsabläufen, sondern eine Strategie, die einen Handlungsspielraum festlegt. Dadurch kann flexibel auf zukünftige Entwicklungen reagiert werden. Der additive Random Walk als einer der Vertreter aus der Familie der stochastischen Prozessen kommt dabei beispielhaft zum Einsatz, da sich damit auch saisonale Schwankungen über die vier Quartale eines Jahres hinweg modellieren lassen. Dieser Vorteil bringt allerdings auch die Einschränkung mit sich, dass sich die Nachfrage relativ stabil entwickeln muss und die Planungsperioden nicht untereinander korreliert sind. Ist das der Fall und das Ergebnis einer Periode nimmt Einfluss auf dessen Folgeperioden, so muss auf andere stochastische Prozesse wie z.B. dem Wiener Prozess oder einer Brownschen Bewegung als Modellgrundlage zurückgegriffen werden.

Durch die kybernetische Sichtweise auf das Planungsproblem wurde das Steuerungsprinzip als treibende Kraft hinter SOPPS ausgewählt. Die Entscheidungsfindung auf Basis einer vergangenheitsbezogenen Sichtweise wird dabei vermieden, der Blick ist immer auf zukünftige Ereignisse gerichtet. Dadurch wird der Anwender nicht durch vergangene Ereignisse in die Irre geführt, in dem er versucht eventuelle Fehlentscheidungen in der Vergangenheit durch risikofreudigere Entscheidungen für die Zukunft wieder zu korrigieren. Durch eine Planung basiert auf Erwartungswerten in Kombination mit der Filtrierung des Wahrscheinlichkeitsraumes, wird im Sinne von Albert Ducrocq der der Zukunft immanente Zufall sukzessive neutralisiert.

Dabei darf nicht unerwähnt bleiben, dass beim SOPPS-Modell von einem symmetrischen Random Walk ausgegangen wurde. Das heißt, sämtliche Trajektorien sind gleich wahrscheinlich. Diese Vereinfachung des Modells deckt nur bedingt mit der Realität. Auch Korrelationseffekte zwischen den einzelnen Perioden blieben unberücksichtigt.

### **4.1.3. Offene Fragen und ungelöste Probleme**

Um die Nachteile der SOPPS aufzugreifen fällt dem Autor als erstes der additive Random Walk auf. Dieser wurde als eines von vielen möglichen Modellen als Basis für die Produktionsplanung implementiert. Obwohl damit die Modellierung saisonaler Effekte möglich wurde, hat dieses Modell auch seine Schwächen. So sind viele mögliche Effekte, die sich auf die Nachfrage auswirken könnten, nicht im Modell enthalten. Allen voran eine geometrische Entwicklung der Nachfrage, welche sich auf Basis der jeweiligen Vorperiode entwickelt. Ebenso blieben korrelative Effekte über mehrere Perioden unbearbeitet.

Ein weiterer Nachteil bedingt durch einer Vereinfachung des Modells ist die Annahme einer symmetrischen Wahrscheinlichkeit über alle möglichen Zustände einer Periode hinweg. Im Sinne von COSO II werden hier Chancen und Risiken mit gleicher Wahrscheinlichkeit bewertet. Eine Risikobewertung ist zwar nicht Teil dieser Arbeit, wäre allerdings nach Ansicht des Autors eine durchaus sinnvolle Ergänzung.

Im in dieser Arbeit beschriebenen Prototyp sind die Trajektorien im Wahrscheinlichkeitsraum auf 10.000 begrenzt. Man ist also in der Lage 10.000 unterschiedliche Endzustände des Planungsbaumes darzustellen. Diese Anzahl rührt einerseits von Performanzüberlegungen, da handelsübliche Computer heutzutage noch nicht über die Kapazitäten verfügen, um eine größere Anzahl zu berechnen, als auch aus didaktischen Gründen der Übersichtlichkeit, da ein Baum mit 10.000 Blättern sich nicht sinnvoll auf einem Bildschirm darstellen lässt. Aus diesen Gründen ist auch die Programmlogik nicht auf mehr Knoten ausgelegt, da die Bezeichnung der Knoten ja auch einem Nummernsystem basiert. Dies erlaubt zwar schon eine recht detaillierte Abbildung der in der Realität möglichen Nachfrage, jedoch nimmt diese Genauigkeit mit zunehmender Produktionsmenge ab. Dieser Umstand ist ein derzeit noch notwendiges Übel, das man hinnehmen muss.

Die Kalibrierung der Nachfrageentwicklung ist im derzeitigen Prototyp auf vier Quartale ausgelegt. Sollte ein Plan mehr als vier Perioden haben, so werden die Werte für das jeweilige Quartal erneut verwendet. Bei einer Weiterentwicklung des Prototyps sollte dieses Manko behoben und eine flexible Eingabe und Speicherung der Kalibrierungsparameter ermöglicht werden.

## 4.2. Zusammenfassung

Diese Arbeit hat eine stochastisch optimale Produktionsplanung und Steuerung auf Basis von Modellen aus der Kybernetik und Stochastik sowie der Berücksichtigung diverser etablierter Standards zum Thema. In der Einleitung wurde die Entwicklung von den Anfängen im MRP-Bereich bis zum heutigen Stand von ERP skizziert. In weiterer Folge wurde das Projekt ERP-Control kurz vorgestellt und auf welchen Konzepten der dazugehörige Prototyp es basiert. Besonders von Interesse war dabei die Verwendung der REA-Ontologie zur Beschreibung von Geschäftsfällen. Diese werden als REA-Entitäten bereits im Datenmodell abgebildet und müssen so nicht erst aus Buchungssätzen aus der Buchhaltung rekonstruiert werden. Diese Verknüpfung hat auch den Vorteil, dass in der strategischen Produktionsplanung Kapazitäten und Ressourceneinsätze nicht berücksichtigt werden mussten, da diese Funktionalität an anderer Stelle abgewickelt wird.

Als Teil davon wurde im Rahmen dieser Arbeit ein Prototyp für eine strategische Produktionsplanung implementiert. Dabei kam ausgehend vom kybernetischen Steuerungsprinzip ein stochastischer Prozess zur Darstellung verschiedener möglicher Nachfragemengen als Zustände innerhalb einer  $\sigma$ -Algebra zum Einsatz. Auf Basis dieser Werte wurde für jeden Zustand eine optimale Produktionsmenge bestimmt, welche auch die Entwicklung der Lagermenge aus der vorhergehenden Periode berücksichtigt. Möglich wurde das auf Basis einer Baumstruktur als Datenmodell, dessen Tiefe und Breite flexibel zur Laufzeit vom Anwender bestimmt werden kann. Bei der Kalibrierung der einzelnen Zustände kam ein additiver stochastischer Prozess zur Anwendung. Dadurch konnten saisonale Unterschiede zwischen den einzelnen Perioden bei der Kalibrierung berücksichtigt werden. Ebenfalls eingeflossen in das Planungsmodell ist die Möglichkeit, technologische Grenzen durch Einschränkungen der Produktionsmengen zu spezifizieren.

Das Ergebnis ist ein Werkzeug zur strategischen Planung und Steuerung der Produktion, welches flexibel auf die Nachfrageänderung innerhalb der spezifizierten Grenzen reagieren kann und dem Anwender dabei hilft, die Produktion optimal der Nachfrage anzupassen und so den Umsatzergebnis zu optimieren.

### 4.3. Blick in die Kristallkugel

Eine strategische Planung auf Basis stochastischer Modelle und unter Einbezugnahme kybernetischer Prinzipien ist zweifelsohne ein sehr spannendes und interessantes Forschungsgebiet, in dem noch sehr viel Potential steckt. Die in Abschnitt 4.1.3 aufgezeigten Schwächen des derzeitigen Prototyps sollten durchaus als Anstoß zu einer zukünftigen Weiterentwicklung verstanden werden.

Bei der Implementierung wurde darauf geachtet, dass das zugrundeliegende Modell in Form des Baumes vollständig von der Befüllung des selbigen durch die Abbildung eines stochastischen Prozesses gekapselt ist. Das ermöglicht es den derzeitigen additiven Random Walk durch einen oder mehrere stochastische Prozesse zu ergänzen um die Flexibilität der SOPPS weiter zu steigern.

Die Entwicklung der Nachfrage kann im momentanen Entwicklungsstand auch nur in einem jener Werte abgebildet werden, die auch im Planungsbaum errechnet wurden. Die Nachfrage lässt sich allerdings nur eingeschränkt durch vom Anwender eingegebenen Erwartungswerten und dessen Variationskoeffizienten beschreiben. Eine Modellanpassung hinsichtlich der Eingabe von realen Nachfragewerten im Zeitverlauf wäre sicherlich eine sinnvolle Ergänzung.

Ein sehr wichtiger Punkt ist eine asymmetrisch Verteilung von Wahrscheinlichkeiten der einzelnen Zustände innerhalb des Wahrscheinlichkeitsraumes. Dieser Punkt wurde im Prototyp nicht umgesetzt. Durch die Möglichkeit verschiedenen Zuständen unterschiedliche Wahrscheinlichkeiten zuzuordnen ließe sich die Realität noch genauer modellieren und eine solche Erweiterung wäre auch im Sinne von COSO II. Dadurch könnte das Risiko einer Fehlplanung bereits in der strategischen Produktionsplanung mit einfließen und dazu beitragen, mögliche Umsatzeinbußen bereits bei der Bestimmung der Produktionsmengen zu berücksichtigen.

Diese drei Beispiele zeigen an sich schon, dass es noch sehr viele Möglichkeiten für Weiterentwicklungen vorhanden sind und man darf gespannt sein, wie sich ERP-Control in Zukunft entwickeln wird.

## 4.4. Persönliches Resümee

Obwohl ich sehr froh und zugegebenermaßen sehr erleichtert bin so weit gekommen zu sein die letzten Zeilen dieser Arbeit schreiben zu können, möchte ich meine Zeit als Diplomand bei Prof. Schwaiger nicht missen. Im Rahmen dieser Arbeit konnte ich einen beachtlichen Wissensschatz über derzeitige ERP-Systeme, deren Funktionsweise und zugrundeliegenden Konzepte und Standards aufbauen.

Besonders spannend war für mich die Idee, REA als Basis für ein BPM-IS zu verwenden, nachdem ich die Ontologie schon aus meiner Bachelorarbeit in einem anderen Kontext kennen gelernt hatte. Für die Darstellung von Geschäftsfällen als Prozess anstatt durch Buchungssätze konnte ich mich schnell begeistern.

Das Thema Stochastik stellte sich als ein durchaus spannendes und interessantes Forschungsgebiet für wirtschaftliche Anwendungen heraus, obwohl ich anfangs meine Schwierigkeiten damit hatte und es eine wirkliche Herausforderung war, in dieses Gebiet einzutauchen. Obwohl ich schon während des Studiums damit in Berührung gekommen bin, habe ich erst als Diplomand die Denkweise richtig verstanden, was mich sicherlich auch für die Zukunft geprägt hat.

Die Implementierung des Prototyps war eine besondere Herausforderung für mich. Datenmodelle zu modellieren und Algorithmen für die Geschäftslogik zu erarbeiten hat ein bisschen etwas von einem Puzzlespiel und einer intellektuellen Herausforderung stelle ich mich gerne. Mein Schwachpunkt war die Visualisierung. Nachdem durchaus anspruchsvolle Technologien in ERP-Control zur Anwendung kommen, musste ich vieles neu erlernen und Zusammenhänge erst verstehen. Und die Fehlersuche ist wohl eines der unangenehmsten Dinge, mit denen ein Programmierer zu tun hat. Oftmals sind es nur Kleinigkeiten, die einen jedoch tagelang beschäftigen und man sich dann ärgert, weil die Lösung dann doch in vielen Fällen entweder einfach oder offensichtlich war.

Obwohl diese Diplomarbeit für mich eine große Herausforderung war, kann ich rückblickend doch überwiegend positive Erfahrungen aus meine Diplomandenzeit mitnehmen. Die Arbeit mit Prof. Schwaiger war sehr angenehm. Er hatte fast

immer Zeit für meine Anliegen und auch eine stoische Geduld und Ausdauer bewiesen, im Speziellen bei den Erläuterungen zur Stochastik in Kombination mit meinem anfänglichen Unverständnis für dieses Thema. Durch diese Arbeit konnte ich mein Verständnis über die interdisziplinären Zusammenhänge in der Wirtschaftsinformatik weiter vertiefen. Im Laufe der vielen Gespräche und Diskussionen war stets Raum für den einen oder anderen Scherz und auch philosophische Diskussionen, die sehr anregend waren und mich zum Nachdenken brachten. Das Engagement und die Einsatzbereitschaft der Erarbeitung der theoretischen Grundlagen war beispielhaft. Dafür möchte ich mich zum Abschluss nochmals für die gute Zusammenarbeit bedanken.

Aus der Arbeit in einem Team mit Mitgliedern unterschiedlicher Fachrichtungen und Nationen konnte ich ebenfalls sehr viele lehrreiche Eindrücke mitnehmen. Ganz besonders die Erarbeitung einer gemeinsamen Sprache war besonders wichtig, da nicht selten aneinander vorbei geredet wurde. Auch das Projektteam war sehr angenehm und es herrschte immer eine produktive Arbeitsatmosphäre und gutes Einvernehmen. Durch die wöchentlichen Meetings konnte ich auch einiges von anderen Diplomanden und ihren Arbeiten lernen, deren Themen in Anbetracht der vielen unterschiedlichen Anwendungsgebieten eines BPM-IS von meiner eigenen Arbeit thematisch doch sehr weit entfernt sind. Besonders bedanken möchte ich mich hier bei Christoph, der mir mit seiner Erfahrung mit dem Projekt und den verwendeten Technologien immer eine Hilfe war, wenn ich nicht weiter wusste.

Ich glaube, der Seiten sind jetzt genug geschrieben. Mir hat die Mitarbeit bei der Implementierung einer ERP-Software sehr gefallen und es gibt noch einiges zu tun. Ich bin schon sehr gespannt auf die weitere Entwicklung von ERP-Control und kann eine Diplomarbeit im Rahmen dieses Projektes vorbehaltlos weiter empfehlen.



## A. Abkürzungsverzeichnis

AEO	Accounting and economic ontology
API	Application Programming Interface
BPM	Business Performance Management
BSC	Balanced Score Card
CIM	Corporate Integrated Management
COSO	Committee of Sponsoring Organisations of the Threadway Commission
DB	Datenbank
ECSI	Enterprise-Control System Integration
EJB	Enterprise Java Beans
ERM	Enterprise Risk Management
ERP	Enterprise Ressource Planning
GP	Geschäftsprozess
GUI	Graphical User Interface
HTML	Hypertext Markup Language
iid	independently and identically distributed
J2EE / Java EE	Java Enterprise Edition
JPA	Java Persistence API
JSF	JavaServer Faces
JSP	JavaServer Pages
KVP	Kontinuierlicher Verbesserungsprozess
MDB	Message-Driven Bean
MIS	Management Information System
MIT	Massachusetts Institute of Technology
MRP	Material Requirements Planning
MRP II	Manufacturing Ressource Planning
MVC	Model-View-Controller
OeBTO	Open-edi Business Transaction Ontology
ORM	Object-relational mapping / O/R-mapping
PC	Personal Computer
PDCA	Plan-Do-Check-Act
POJO	Plain Old Java Object
SFSB	Stateful Session Bean
SLSB	Stateless Session Bean
SOA	Serviceorientierte Architektur
SOPPS	Stochastisch optimale Produktionsplanung und –steuerung
(G)UI	(Graphical) User Interface
XML	Extensible Markup Language

Tabelle 5: Abkürzungsverzeichnis

## B. Notationsverzeichnis

$k_{v,p}$	Variable Produktionskosten
$X_{L,t}$	Lagermenge
$k_{v,L,t}$	Lagerkosten
$K_F$	Fixkosten
$X_{P,t}$	Produktionsmenge
$\tilde{X}_{Abs,t}$	Absatzmenge
$\tilde{X}_{N,t}$	Nachfragemenge
$p$	Verkaufspreis
$\tilde{B}E$	Betriebsergebnis
$\tilde{K}$	Gesamtkosten
$\tilde{U}MS$	Umsatz
$db_t$	Deckungsbeitrag
$\Omega$	Stichprobenmenge innerhalb des Wahrscheinlichkeitsraumes
$A$	System, welches durch eine Ansammlung von $\Omega$ beschrieben wird ( $\sigma$ -Algebra)
$E$	Messbare Ereignisse innerhalb der $\sigma$ -Algebra
$s_{t,i}$	Zustand i zum Zeitpunkt t
$a_t$	Ausgangswert für den additiven Random Walk zum Zeitpunkt t

Tabelle 6: Notation und Symbole

## C. Abbildungsverzeichnis

Abbildung 1: Drei-Schichten-Architektur .....	10
Abbildung 2: ERP-Control .....	12
Abbildung 3: ERP-Control Konzept .....	13
Abbildung 4: REA-Ontologie .....	17
Abbildung 5: Kybernetische Managementmodelle - Unterscheidung von Regelungs- und Steuerungsmodell .....	21
Abbildung 6: Funktionale Hierarchie .....	24
Abbildung 7: Funktionales Unternehmensführungs-/Leitsystemmodell .....	25
Abbildung 8: Materialmodell .....	27
Abbildung 9: Schematischer Zusammenhang zwischen Planung und Steuerung .....	28
Abbildung 10: REA ergänzt um Business Events .....	30
Abbildung 11: Business Transaction Model mit gebündelten Commitments .....	30
Abbildung 12: COSO-Würfel .....	32
Abbildung 13: Balanced Score Card .....	36
Abbildung 14: Nummerierung der Baumknoten .....	51
Abbildung 15: Problem multipler Zustandsvariablen .....	53
Abbildung 16: Ablauf der Baumgenerierung .....	67
Abbildung 17: Geschäftsprozess der Plangenerierung .....	75
Abbildung 18: Ablauf Plangenerierung .....	76
Abbildung 19: Plan benennen .....	76
Abbildung 20: Produktauswahl .....	77
Abbildung 21: Modellparameter spezifizieren .....	77
Abbildung 22: Produktionsparameter spezifizieren .....	78
Abbildung 23: Fehlermeldung bei nicht plausiblen Werten .....	80
Abbildung 24: Plan speichern .....	80
Abbildung 25: Bestätigung des gespeicherten Plans .....	81
Abbildung 26: Geschäftsprozess das Planmodifikation .....	81
Abbildung 27: Geschäftsprozess Genehmigung des Plans .....	82
Abbildung 28: Ablauf Plangenehmigung .....	82
Abbildung 29: Auswahl des Planes .....	83
Abbildung 30: Übersicht und Bestätigung des Plans .....	83
Abbildung 31: Geschäftsprozess Simulation des Plans .....	84
Abbildung 32: Ablauf der Simulation .....	84
Abbildung 33: Übersichtstabelle des Planfortschritts .....	84
Abbildung 34: Darstellung und Auswahl der Baumknoten .....	85

## D. Tabellenverzeichnis

Tabelle 1: Attribute von Materialloseigenschaft.....	28
Tabelle 2: Zustandsraum .....	44
Tabelle 3: Partitionierung des Wahrscheinlichkeitsraumes .....	46
Tabelle 4: Speicherbedarf der Baumstruktur.....	52
Tabelle 5: Abkürzungsverzeichnis .....	a
Tabelle 6: Notation und Symbole .....	b

## E. Formelverzeichnis

Formel 1: Definition Lagermenge .....	39
Formel 2: Definition Absatzmenge .....	40
Formel 3: Gesamtkosten.....	40
Formel 4: Umsatz.....	40
Formel 5: Herleitung Umsatzergebnis.....	41
Formel 6: Definition $\sigma$ -Algebra .....	42
Formel 7: Zustand im geometrischen RW .....	44
Formel 8: Zustand im additiven RW .....	44
Formel 9: Fakultät .....	47
Formel 10: Berechnung der Nachfrage .....	69

## F. Codeverzeichnis

Codefragment 1: Klasse ProductionPlanModelEntity .....	62
Codefragment 2: Klasse Node .....	64
Codefragment 3: Code zur Generierung des Baumes .....	66
Codefragment 4: Nachfrage in Node einfügen .....	69
Codefragment 5: Ermittlung des Umsatzergebnisses .....	71
Codefragment 6: Bestimmung der optimalen Produktionsmenge .....	71
Codefragment 7: Ermittlung des erwarteten Umsatzergebnisses .....	72
Codefragment 8: Berechnung des intertemporalen Umsatzergebnisses.....	73

## G. Literaturverzeichnis

1. **Knöpfel, Holger und Löwe, Matthias.** *Stochastik - Struktur im Zufall.* München, Wien : Oldenbourg Wissenschaftsverlag GmbH, 2007. ISBN 978-3-486-58448-6.
2. **Burns, Tom.** *Auditing MRP systems.* [Hrsg.] ACM. ACM SIGSAC Review. Juni 1993, Volume 11, Issue 3, S. 2-13.
3. **Seiringer, Wolfgang.** *Planung und Unterstützung der Entscheidungsfindung für die Auswahl eines ERP-Systems eines mittelständischen Unternehmens.* [Masterarbeit (TU Wien)]. Wien, Wien, Österreich : TU Wien, 2009.
4. **Buhl, Dipl.-Ing.(FH) Matthias.** *Produktionsplanung und -steuerung in mittelständischen Unternehmen unter besonderer Berücksichtigung von Enterprise Resource Planning Systems.* [Dissertation (TU Wien)]. Wien : TU Wien, 2008.
5. **Monk, Ellen.** *Concepts in Enterprise Resource Planning.* [Hrsg.] Course Technology Cengage Learning. 3. Boston : s.n., 2009. ISBN-10: 1-4239-0179-7.
6. **Hesseler, Martin und Görtz, Marcus.** *Basiswissen ERP-Systeme - Auswahl, Einführung & Einsatz betriebswirtschaftlicher Standardsoftware.* Witten : W3L GmbH, 2007. ISBN 978-3-937137-38-4.
7. **Oreilley, Tim.** *What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software.* [Hrsg.] O'Reilly Media. Communications & Strategies. 2007, 65. Ausgabe.
8. **Mathas, Christoph.** *SOA intern: Praxiswissen zu serviceorientierten IT-Systemen.* München : Hanser Fachbuch, 2007. ISBN-10: 3446411895.
9. **ISO/IEC JTC 1/SC 32/WG1.** *ISO/IEC 15944-4:2006 Information technology - Business Operational View -- Part 4: Business transaction scenarios - Accounting and economic ontology.* [Hrsg.] ISO. [Standard]. 17. 01 2006. ISO/IEC 15944-4.

10. **Wiener, Norbert.** *Kybernetik: Regelung und Nachrichtenübertragung im Lebewesen und in der Maschine.* Düsseldorf : ECON Verlag, 1992.  
ISBN 3-430-19652-3.

11. **Rothmayer, Rolf.** *Kybernetik : wir kommen ohne sie nicht mehr aus: wie die Steuer- und Regelungsvorgänge in Biologie, Technik, Physik, Medizin und Soziologie funktionieren.* Wien : hpt-Verlagsgesellschaft, 1988.  
ISBN 3-85128-015-6.

12. **Flechtner, Hans Joachim.** *Grundbegriffe der Kybernetik : eine Einführung.* 3. Auflage. Stuttgart : Wissenschaftliche Verlagsgesellschaft m.b.H., 1968.

13. **Schwaiger, Walter S.A.** *Integration des Risiko-Managements in das Führungssystem von Banken.* [Hrsg.] Roland Eller. re|peat - Jahrbuch Treasury und Private Banking 2011. 2010, 1.

14. **Austrian Standards Institute.** Austrian Standards Institute. [Online]  
[Zitat vom: 14. April 2010.] <http://www.as-institute.at/development/normen-entwickeln/grundlagen/was-ist-eine-norm/>.

15. **DIN Deutsches Institut für Normung e.V.** *DIN EN 62264-1 Integration von Unternehmensführungs- und Leitsystemen – Teil 1: Modelle und Terminologie (IEC 62264-1:2003) Deutsche Fassung EN 62264-1:2008.* Berlin : Beuth Verlag GmbH, 2008. Bd. 1. ICS 01.040.35; 35.240.50.

16. —. *DIN EN 62264-2 Integration von Unternehmensführungs- und Leitsystemen – Teil 2: Attribute des Objektmodells (IEC 62264-2:2004); Englische Fassung EN 62264-2:2008.* Berlin : Beuth Verlag GmbH, 2008. Bd. 2. ICS 35.240.50.

17. —. *DIN EN 62264-3 Integration von Unternehmens-EDV und Leitsystemen – Teil 3: Aktivitätsmodelle für das operative Produktionsmanagement (IEC 62264-3:2007); Deutsche Fassung EN 62264-3:2007, Text Englisch.* Berlin : Beuth Verlag GmbH, 2007. Bd. 3. ICS 35.240.01.

18. **Comittee of Sponsoring Organizations of the Threadway Commission; Institut für Interne Revision Österreich.** *Enterprise Risk Management - Integrated Framework. Deutsche Fassung.* Wien : s.n., 2009.  
ISBN 978-3-9502676-1-7.

19. **Schwab, Adolf J.** *Managementwissen für Ingenieure*. 4. Auflage. Berlin : Springer-Verlag, 2008. ISBN 978-3-540-78409-8.
20. **Kreuzer, Christian.** *BWL kompakt*. 2. Auflage. Wien : Linde Verlag Wien GmbH, 2007. ISBN 978-3-7143-0103-8.
21. **Mesarovic, M. D., Macko, D. und Takahara, Y.** *Theory of Hierarchical, Multilevel, Systems*. New York : Academic Press, 1970. LCCN 79-117116.
22. **Schwaiger, Walter S.A.** *Controlling unter Unsicherheit*. [Hrsg.] TU Wien Institut für Managementwissenschaften. Wien : s.n., Oktober 2009. Skriptum zur Vorlesung.
23. **Hübner, Gerhard.** *Stochastik*. Braunschweig/Wiesbaden : Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, 1996. ISBN 3-528-05443-3.
24. **Schlittgen, Rainer und Streitberg, Bernd H. J.** *Zeitreihenanalyse*. 9. Auflage. München, Wien : Oldenbourg Wissenschaftsverlag GmbH, 2001. ISBN 3-486-25725-0.
25. **Bauer, Heinz.** *Wahrscheinlichkeitstheorie*. 5. Auflage. Berlin : Walter de Gruyter GmbH und CO.KG, 2002. ISBN 978-3-11-017236-2.
26. **Bertsekas, Dimitri P. und Shreve, Steven E.** *Stochastic Optimal Control*. New York : Academic Press Inc., 1978. ISBN 0-12-093260-1.
27. **Papageorgiou, Markos.** *Optimierung*. München : Oldenbourg Verlag München, 1991. ISBN 3-486-21799-2.
28. **Rupp, Heiko W.** *EJB 3 für Umsteiger*. 1. Auflage. Heidelberg : dpunkt.verlag GmbH, 2007. ISBN 978-3-89864-429-7.
29. **Klute, Rainer.** *Mit neuem Gesicht - Java Server Faces vereinfachen Webentwicklung*. iX Magazin für professionelle Informationstechnik. 11/2004, S. 128 ff.

30. **Marinschek, Martin, Kurz, Michael und Müllan, Gerald.** *JavaServer Faces 2.0 : Grundlagen und erweiterte Konzepte*. 2. Auflage. Heidelberg : dpunkt.Verlag GmbH, 2010. ISBN 978-3-89864-606-2.
31. **Yuan, Michael Juntao und Heute, Thomas.** *JBoss Seam Enterprise-Webanwendungen mit Java EE - einfacher und leistungsstärker*. 1. Auflage. Heidelberg : mitp-Verlag, 2008. ISBN 978-3-8266-1797-3.
32. **Instrument Society of America.** *ANSI/ISA–95.00.02–2001 Enterprise-Control System Integration Part 2: Object Model Attributes*. Research Triangle Park : Instrument Society of America, 2001. Bd. 2. ISBN 1-55617-773-9.
33. —. *ANSI/ISA–S95.00.01–2000 Enterprise-Control System Integration Part 1: Models and Terminology*. Research Triangle Park : Instrument Society of America, 2000. Bd. 1. ISBN 1-55617-727-5.
34. **Stepan, Adolf und Fischer, Edwin O.** *Betriebswirtschaftliche Optimierung - Einführung in die quantitative Betriebswirtschaftslehre*. 8. München : Oldenbourg Verlag München, 2009. ISBN 978-3-486-58781-4.
35. **Schierenbeck, Henner und Wöhle, Claudia B.** *Grundzüge der Betriebswirtschaftslehre*. 17. München : Oldenbourg Verlag München, 2008. ISBN 978-3-486-58772-2.
36. **McCarthy, William E.** *The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment*. [Hrsg.] American Accounting Association. *The Accounting Review*. Vol. LVII, Juli 1982, 3.
37. **Hruby, Pavel.** *Model-Driven Design Using Business Patterns*. Berlin : Springer-Verlag, 2006. ISBN 3-540-30154-2.
38. **Amschl, Raphael, et al.** *Web Worksheet Editor*. [Bakkalaureatsarbeit (Universität Wien)]. Wien : Universität Wien, Juli 2007.
39. **UNECE. UN/CEFACT.** [Online] 22. September 2003. [Zitat vom: 09. Mai 2007.] [http://www.unece.org/cefact/umm/UMM\\_userguide\\_220606.pdf](http://www.unece.org/cefact/umm/UMM_userguide_220606.pdf).



40. **Instrumentation, Systems, and Automation Society.** *ANSI/ISA-95.00.03-2005 Enterprise-Control System Integration Part3: Activity Models of Manufacturing Operations Management.* Research Triangle Park : Instrumentation, Systems, and Automation Society, 2005. Bd. 3. ISBN 1-55617-955-3.
41. **Kurbel, Prof. Dr. Karl.** *Produktionsplanung und -steuerung im Enterprise Ressource Planning und Supply Chain Management.* 6. Auflage München : Oldenbourg Wissenschaftsverlag GmbH, 2005. ISBN 3-486-57578-3.
42. **Carnogursky, Martin.** *Evaluation and Redesign of ERP systems: a methodological approach.* [Hrsg.] TU Wien. [Diplomarbeit]. Wien : s.n., 9. März 2009.
43. **Deming, William Edwards.** *Out of the Crisis.* Cambridge : MIT Press, 2000. ISBN 0-262-54115-7.
44. **Eberling, Werner und Lessner, Jan.** *Enterprise Java Beans 3 Das EJB3-Praxisbuch für Ein- und Umsteiger.* München, Wien : Carl Hanser Verlag, 2007. ISBN 978-3-446-41085-5.
45. **Marinschek, Martin, Schnabl, Andrea und Müllan, Gerald.** *JSF @ Work.* 1. Auflage. Heidelberg : dpunkt.verlag GmbH, 2007. ISBN-10: 2-89864-401-4.