

**Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

**Planung von Produktausläufen in Supply Chains
Entwicklung eines Software-Tools
und softwaretechnische Auswertung**

Bachelorarbeit

im Studiengang Informatik

von

Lars Wolter

Prüfer: Prof. Dr. Kurt Schneider

Zweitprüfer: Prof. Dr.-Ing. habil. Peter Nyhuis

Betreuer: M. Sc. Sebastian Meyer

Hannover, 12.04.2010

Danksagung

Hiermit möchte ich mich bei allen Personen, die zu dieser Arbeit beigetragen und mich unterstützt haben, bedanken. Besonderer Dank geht an die Prüfer Prof. Dr. Kurt Schneider und Prof. Dr. Peter Nyhuis sowie an Sebastian Meyer und Frauke Hertrampf für die Betreuung und Hilfsbereitschaft während der Arbeit.

Inhaltsverzeichnis

1 Kurze Zusammenfassung	4
2 Gliederung	5
3 Motivation: Warum ein Software-Tool?	6
4 Einführung in das Thema und Begriffserläuterungen	8
4.1 Produktausläufe.....	8
4.2 Lieferketten.....	9
4.3 Vorgehen bei der Planung von Produktausläufen.....	10
5 Domänen erkundung und Anforderungserhebung	13
5.1 Chronologischer Ablauf.....	13
5.2 Zusammenfassung und Auswertung der Erfahrungen.....	15
6 Anforderungen an das Software-Tool	18
7 Der ALEKS Demonstrator: Aufbau und Struktur	21
7.1 Das Produkt und seine Lieferkette.....	21
7.2 Dateioperationen.....	22
7.3 Diagramme.....	23
7.4 Der Losgrößenalgorithmus.....	24
7.5 Die grafische Benutzerschnittstelle (GUI).....	25
8 Ausblick und kritische Würdigung	27
9 Literaturverzeichnis	29
10 Abbildungs- und Tabellenverzeichnis	30
11 Selbstständigkeitserklärung	31

Anlage

CD mit der Software, dem Quellcode und der schriftlichen Ausarbeitung

1 Kurze Zusammenfassung

Das im Laufe dieser Arbeit entwickelte Software-Tool dient der Visualisierung von Lieferketten und Produktionszusammenhängen beliebiger Produkte - das Hauptaugenmerk liegt hier aber auf Produkten der Elektronikindustrie. Es soll seinem Anwender außerdem bei der Berechnung der kostengünstigsten Produktionskonfigurationen behilflich sein und die Ausnutzung von Einsparpotenzialen ermöglichen.

Da diese Arbeit in einer fremden Domäne stattfindet, hat deren Erkundung einen sehr hohen Stellenwert. Zunächst ist es also nötig, sich benötigtes Wissen im Bereich der Produktionslogistik anzueignen, da dies für das Verständnis der zu implementierenden Funktionen unabdingbar ist. Im Anschluss folgen das Verständnis der Arbeitsweise in diesem Fachbereich sowie die Erhebung der zu implementierenden Anforderungen und Funktionen in persönlichen Gesprächen. Nach einer ersten Entwicklungsphase wird dann ein veranschaulichender Prototyp vorgestellt, an dem von diesem Zeitpunkt an in Abstimmung mit dem betreffenden Institut fortwährend die gewünschten Änderungen und Erweiterungen vorgenommen werden.

Im Anschluss folgen neben der Vorstellung des entstandenen Software-Tools und der Erläuterung seines Aufbaus eine softwaretechnische Auswertung sowie die Beleuchtung aufgetretener Probleme und die Ausführung der aus dieser Arbeit gezogenen Schlüsse. Dies soll in dieser schriftlichen Ausarbeitung geschehen.

2 Gliederung

Diese schriftliche Ausarbeitung befasst sich im folgenden dritten Kapitel mit der Motivation dieser Arbeit. Es werden die Umstände dargelegt, die zu dem Wunsch nach einem Software-Tool, das an die speziellen Anforderungen angepasst ist, führten und die Notwendigkeit anhand des bisherigen unbefriedigenden Zustands erläutert.

Das vierte Kapitel soll dem Leser das nötige Fachwissen im Bereich der Produktionslogistik näherbringen. Das Verständnis der in diesem Kapitel erläuterten Grundbegriffe ist vonnöten, um die darauffolgenden Kapitel nachvollziehen zu können.

Kapitel 5 beschreibt anschließend das Vorgehen bei der Erkundung der fremden Domäne anhand eines chronologischen Ablaufs. Es werden die Probleme, die es bei der Zusammenarbeit gegeben hat, dargelegt und in einem anschließenden Abschnitt Schlüsse gezogen und mögliche Quellen für Missverständnisse bei der Kommunikation zweier einander fremden Fachbereiche herausgearbeitet.

Das sechste Kapitel befasst sich mit der Ausarbeitung der schlussendlich erhobenen Anforderungen an das Software-Tool. Es liefert somit einen Überblick über die realisierten Funktionen.

Kapitel 7 gewährt einen tieferen Einblick in die Struktur sowie den Aufbau des Programms und zeigt, auf welche Weise die geforderten Funktionen umgesetzt wurden. Die Klassen der einzelnen Komponenten werden hier auch mit Hilfe einiger UML-Diagramme bezüglich ihrer Funktion erläutert und ihre Beziehungen untereinander dargelegt.

Das achte Kapitel liefert schließlich einen Überblick über eine mögliche zukünftige Weiterentwicklung des Software-Tools und gibt ein paar anregende Beispiele, wie der Funktionsumfang erweitert und der Nutzen für den Anwender somit vergrößert werden könnte.

3 Motivation: Warum ein Software-Tool?

Die Arbeit an dem Programm erfolgt in Zusammenarbeit mit dem Institut für integrierte Produktion Hannover (IPH), das im Bereich der Logistik an möglichst optimalen Lieferkettenkonfigurationen und der Berechnung von entsprechend kostengünstigen Produktausläufen forscht. Dafür ist es nötig die dem jeweiligen Produkt zugrunde liegende Lieferkette möglichst realitätsnah abbilden und die Produktionszusammenhänge ihrer einzelnen Stufen visualisieren zu können. Auf dieser Grundlage können dann gewisse Berechnungen durchgeführt werden, um den gesamten Produktionsvorgang zu optimieren und Einsparpotenziale vor allem beim Auslauf des Produkts ausnutzen zu können.

Das IPH nutzte für diese Belange bisher eine selbst erstellte Excel-Tabelle, die den Anforderungen allerdings schon bald nicht mehr gerecht wurde. Es mussten hier pro Lieferkettenstufe bis zu drei Tabellenblätter erstellt werden, welche die teilweise riesigen Mengen an Daten des auf diese Weise in Wareneingang, Produktion und Warenausgang unterteilten Zulieferers verwalteten. Dieser enorme Umfang führte auch dazu, dass die ebenfalls auf diesen Tabellenblättern untergebrachten Diagramme, die eigentlich der Veranschaulichung und dem besseren Verständnis dienen sollten, teilweise erst umständlich gesucht werden mussten. Darüber hinaus führte die Konfiguration des Excel-Dokuments auf ein neues Produkt zu massiven Änderungen an allen Tabellenblättern. Neben der Aktualisierung und Erweiterung der unzähligen Tabellen und Parameter bedurften auch Formeln, Verweise und Diagramme umständlicher Anpassungen.

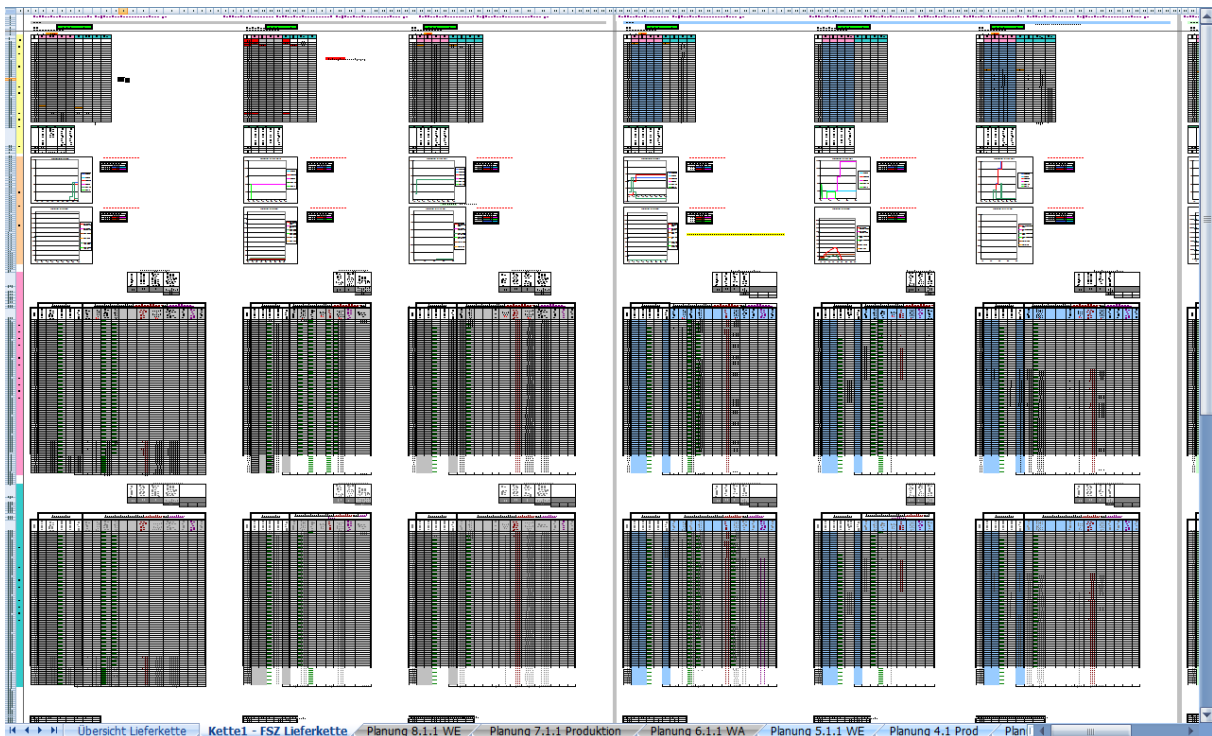


Abbildung 1: Ausschnitt aus der bisher verwendeten Excel-Tabelle (Zoomfaktor 10%)

Der Versuch, mit Excel eine Problemstellung dieser Dimension zu lösen, stieß an dieser Stelle aufgrund der Vielfältigkeit von Lieferketten, der variablen Dauern von Produktlebenszyklen und der unterschiedlichen Mengen von Kundenbestellungen an seine Grenzen. Wie Abbildung 1 anschaulich anhand eines herausgezoomten Ausschnitts aus dem Dokument zeigt, war das Ergebnis dieses Versuchs eine wenig praktikable mehrseitige Ansammlung großer Tabellenkonstrukte, die schlecht zu warten waren und die gewünschte Übersichtlichkeit und Benutzerfreundlichkeit nicht gewährleisten konnten.

Ein neu entwickeltes Software-Tool hingegen kann die Daten dynamisch verwalten und minimiert somit den Arbeitsaufwand für den Benutzer. Es lässt sich genau an die Bedürfnisse des Anwenders anpassen; vor allem aber bietet es die Möglichkeit einer einfachen Erweiterung im Laufe der Zeit. Die freie Anpassbarkeit der grafischen Benutzeroberfläche spricht in diesem Fall ebenfalls für eine eigens erstellte Software, da so auch eine zeitraubende Einarbeitung neuer Nutzer in das Programm, wie sie bei Nutzung der Excel-Tabelle nötig gewesen wäre, entfällt.

Das im Laufe dieser Bachelorarbeit erstellte Software-Tool soll vor allem eine demonstrierende Funktion erfüllen, also durch anschauliche Visualisierungen verschiedene Parameterkonfigurationen vergleichbar machen und so die zukünftige Forschung des IPH unterstützen.

4 Einführung in das Thema und Begriffserläuterungen

4.1 Produktausläufe

Wie Abbildung 2 zeigt, durchläuft ein Produkt nach seiner Markteinführung verschiedene Phasen, die sich durch die Absatzzahlen und Kosten des Produkts bestimmen lassen [Hom09].

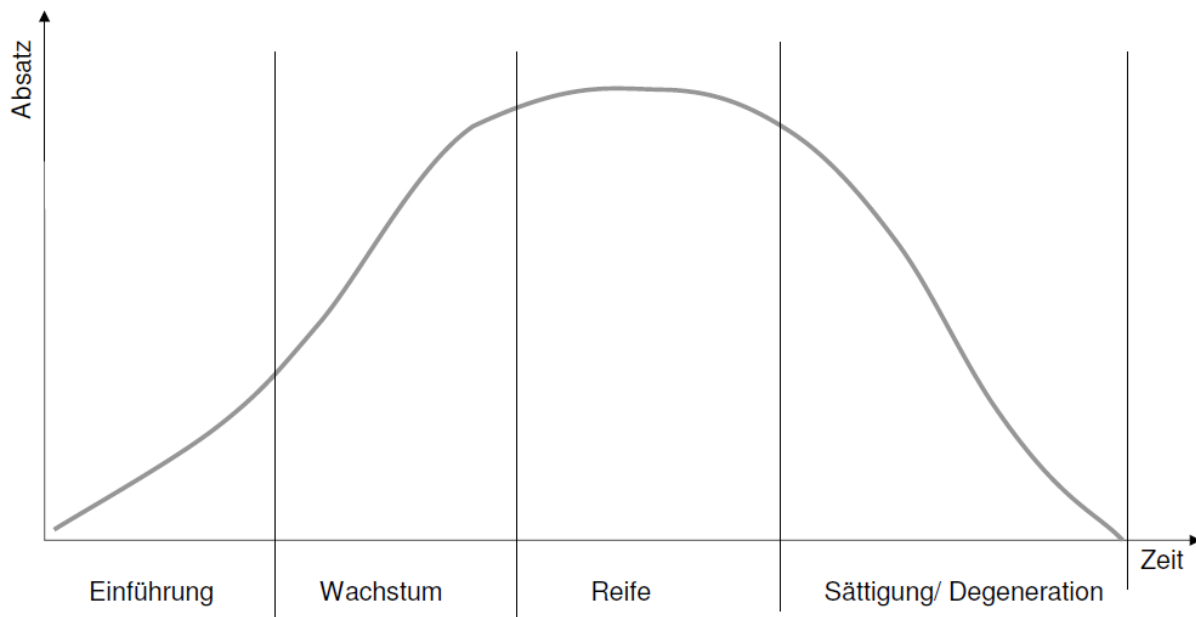


Abbildung 2: Absatzverlauf im Produktlebenszyklus

Es beginnt in der **Markteinführungsphase**, die noch mit hohen Markteinführungskosten (Werbung, Kundenakquise) und sehr geringen Erlösen verbunden ist. Das Produkt ist noch weitgehend unbekannt, der Absatz gering und das produzierende Unternehmen muss Verluste in Kauf nehmen. Während im Laufe der Zeit die Absatzzahlen langsam ansteigen, wechselt das Produkt in die **Wachstumsphase**. Die Werbekosten sind zwar immer noch hoch aber dennoch sind in dieser Phase die ersten kleinen Gewinne zu verzeichnen. Die lohnendste Phase ist die anschließende **Reifephase**. Das Produkt ist nun etabliert, die Werbekosten daher gering, und es erzeugt hier die höchsten Gewinne. Im Allgemeinen wird daher versucht, diese Phase mit Marketingmaßnahmen so weit es gelingt in die Länge zu ziehen. Jedoch kommt es bei fast jedem Produkt früher oder später zu einem Abfall der Absatzzahlen - es tritt also in die **Sättigungs- bzw. Degenerationsphase** ein. Sobald es am Ende keine oder nur noch geringe Gewinne abwirft, lässt man es für gewöhnlich auslaufen, d. h. man nimmt es vom Markt (End-of-Life), um Verluste zu vermeiden.

Gründe für einen solchen Absatzrückgang, der schließlich zu einem Auslauf des Produkts führt, können eine zunehmende Deckung des Bedarfs, eine stärker gewordene Konkurrenz oder etwa technischer Fortschritt sein. Allerdings gibt es auch unternehmensinterne Gründe dafür ein Produkt auslaufen zu lassen, die nicht oder zumindest nicht nur in den Absatzzahlen

begründet liegen. Bereinigungen des Produktportfolios oder auch das Re-Design eines Produkts können hier die Ursache sein.

Diese oben beschriebenen Produktlebenszyklen haben sich durch technischen Fortschritt und Schnelligkeit in der Vergangenheit immer mehr verkürzt, wodurch es auch immer häufiger zu Produktausläufen kommt. Diese beziehen sich allerdings nicht nur auf Endprodukte sondern auch Baugruppen und Einzelteile werden häufig durch neuere Versionen ersetzt, ohne dass dies vom Endkunden bemerkt wird.

Unzureichende oder sogar fehlende Auslaufplanung führt dabei meist entweder zu hohen Restbeständen und den damit verbundenen Lager- oder Entsorgungskosten oder zu Fehlteilen bei der Ersatzteilversorgung und der damit verbundenen kostenintensiven Nachbeschaffung. Der Produktauslauf birgt somit ein hohes Kosteneinsparpotenzial und macht eine intensive Planung notwendig – eine Planung, die lieferkettenübergreifend erfolgen sollte, da eine isolierte Auslaufplanung für ein einzelnes Unternehmen nicht zwangsläufig für die gesamte Lieferkette optimal ist und bei Partnern sogar zu erhöhten Kosten führen könnte.

Eine zusätzliche Herausforderung hierbei ist, dass der Auslauf eines Endprodukts keinesfalls zwingend den Auslauf aller Einzelteile auslöst. Eventuell werden einige Einzelteile auch in anderen Erzeugnissen verbaut und müssen weiterproduziert werden. Insbesondere in der Elektronikindustrie sind die Produkte einerseits durch eine hohe Komplexität gekennzeichnet, bestehen also aus einer Vielzahl von Einzelteilen, von denen viele auch in anderen Produkten verbaut werden, und andererseits besonders vom technischen Fortschritt und der damit verbundenen Schnelligkeit betroffen.

4.2 Lieferketten

Eine Lieferkette ist eine meist baumartige Struktur, die den Weg der Einzelteile bis zum Endprodukt darstellt. Ein anschauliches Beispiel hierfür liefert Abbildung 3. Der letzte Knoten stellt dabei die Stufe vor der Auslieferung an den Kunden dar. Von hier aus wird nun nach hinten verzweigend die Herstellung eines jeden für die Produktherstellung benötigten Einzelteils durch einen darunter liegenden Knoten abgebildet, die Einzelteile dieser Einzelteile wiederum durch weitere Knoten.

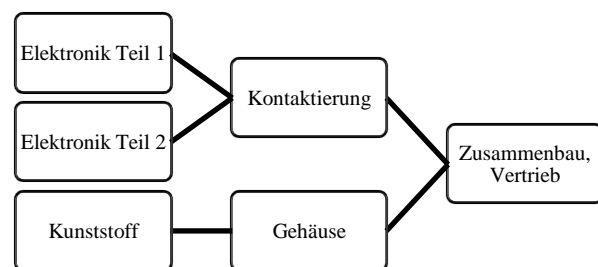


Abbildung 3: Beispiel einer Lieferkette

Die Einzelteile dieser Einzelteile wiederum durch weitere Knoten.

Lieferketten können dadurch bei einfachen Produkten sehr einfach gestaltete Gebilde mit wenigen Knoten sein - je nach Komplexität des Erzeugnisses kann ihr Verzweigungsgrad aber auch stark ansteigen. Dabei ist es allerdings nicht immer nötig, die komplette Lieferkette bis

ins kleinste Detail abzubilden. In manchen Fällen kann es hilfreich sein, für die Planung irrelevante Bereiche der Einfachheit halber auszublenden.

Jede Stufe einer Lieferkette sieht ihre direkten Vorgänger als Zulieferer und ihren Nachfolger als Kunden an. Hierdurch soll die lieferketteninterne Zuverlässigkeit der Warenlieferungen erhöht werden, da die termingerechte Befriedigung eines Kundenbedarfs für ein Unternehmen oberste Priorität haben sollte.

Jede Stufe besitzt außerdem einige bestimmte Parameter und Eigenschaften, die für lieferkettenübergreifende Berechnungen wichtig sein können. Diese Parameter sind bedeutende Faktoren bei der kostenoptimierenden Planung der Produktausläufe und müssen bei Berechnungen berücksichtigt werden.

Tabelle 1 nennt und beschreibt die in diesem Fall wichtigen Parameter.

Parameter	Beschreibung
Produktionsdauer	Dauer der in dieser Stufe vorgenommenen Arbeitsschritte zwischen Eintreffen aller Rohstoffe und der Fertigstellung des (Zwischen-)Produkts
Lieferzeit	Dauer des Liefervorgangs zur nachfolgenden Stufe bzw. zum Endkunden
Gewichtung	Menge mit der dieses Zwischenprodukt in das Folgeerzeugnis einfließt
Bestellkosten	Alle Kosten, die mit der Bestellung dieses Erzeugnisses einhergehen
Stückkosten	Kosten pro Einheit dieses Erzeugnisses
Rüstkosten	Kosten, die einmalig beim Start der Produktion anfallen z.B. Ingangsetzung der Maschinen

Tabelle 1: Parameter einer Lieferkettenstufe

4.3 Vorgehen bei der Planung von Produktausläufen

Die Berechnungsgrundlage einer Produktauslaufplanung bildet in erster Linie die gesamte **Kundenbedarfsmenge**, also eine Menge von über den gesamten Produktlebenszyklus verteilten Zeitpunkten, zu denen der Kunde bestimmte Mengen des Produkts beziehen möchte. Dieser Kundenbedarfsverlauf ist zu Beginn entweder ganz oder teilweise durch zuvor abgeschlossene Verträge festgelegt oder wird anhand bereits gemachter Erfahrungen von den jeweiligen liefernden Unternehmen angenommen. Ist der Verlauf des Kundenbedarfs nicht sicher, so muss allerdings eine gewisse mögliche Abweichung sowohl nach oben als auch nach unten mit in die Risikoberechnung einfließen. Je größer der unsichere Anteil der Kundenbedarfsmenge ist desto größer sind auch die mögliche Abweichung und das damit verbundene Risiko. Die im schlechtesten Fall anfallenden Mehrkosten müssen berechnet werden und mindern als **Wagniskosten** den Gewinn des Unternehmens.

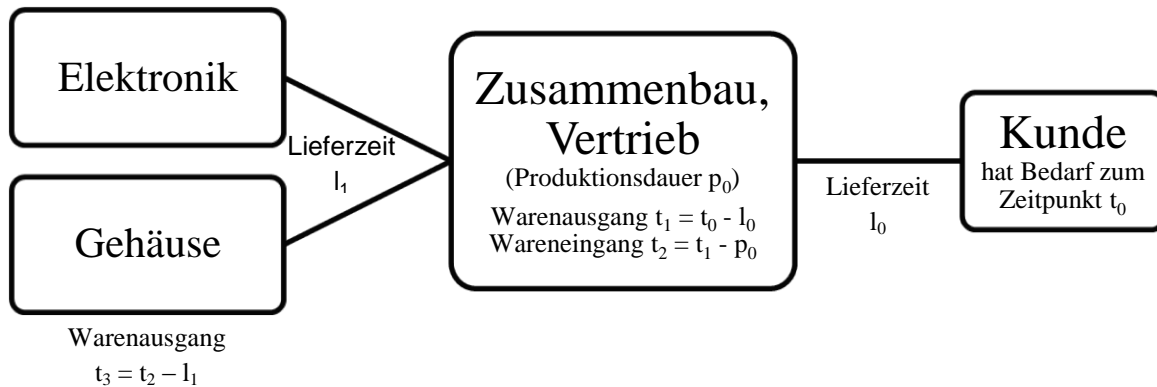


Abbildung 4: Berechnung der Sollfortschrittszahlen in einer Lieferkette

Aus diesem Kundenbedarf berechnen sich nun, wie in Abbildung 4 illustriert, die sogenannten **Sollfortschrittszahlen** für jede einzelne Stufe der Lieferkette:

Zieht man von den einzelnen Kundenbedarfsterminen die jeweilige Lieferzeit zum Kunden ab, so erhält man die **Sollabgangszahlen** für die Wurzelstufe. Dies sind die spätmöglichen Zeitpunkte für die Auslieferung an den Kunden. Aus diesen Zeitpunkten wiederum entstehen abzüglich der Produktionsdauer die **Sollzugangszahlen**, also die Zeitpunkte, zu denen die Produktion jeweils spätestens anlaufen muss und die für die Produktion benötigten Einzelteile im Wareneingangslager vorhanden sein müssen. Daraus können nun wieder die Sollzugangs- und -abgangszahlen der vorherigen Stufen berechnet werden. Das Vorgehen ist in etwa vergleichbar mit den unter anderem aus der Software-Technik bekannten Netzplänen. Dort werden bei der zeitlichen Planung eines Projekt ebenfalls vom Zeitpunkt der geplanten Fertigstellung ausgehend die spätestmöglichen Startzeitpunkte der einzelnen Vorstufen anhand ihrer Arbeitszeit errechnet. Die Lieferkette ist mit ihrer sich nach hinten verbreiternden Baumstruktur lediglich einfacher gestaltet als ein Netz.

Bei diesem Vorgang bildet sich zwischen den einzelnen Stufen der Lieferkette ein sogenanntes **virtuelles Lager**. Die Zwischenprodukte werden direkt nach ihrer Produktion diesem virtuellen Lager zugerechnet, sie verlassen es erst wieder, sobald sie in den Produktionsdurchlauf der nachfolgenden Stufe eintreten. Dieses Lager fasst also mit anderen Worten den Aufenthalt der Waren nach ihrer Fertigstellung im Ausgangslager der produzierenden Stufe und im Eingangslager der nachfolgenden Stufe sowie die Zeit, die dazwischen für den Liefervorgang benötigt wird, zu einer Art Zwischenstufe zusammen. Der Abgangsgraph einer Lieferkettenstufe ist somit gleich dem Eingangsgraphen des virtuellen Lagers. Der Eingangsgraph der nachfolgenden Stufe wiederum stellt den Abgangsgraphen des virtuellen Lagers dar. Aus der Differenz der beiden Zahlenreihen lässt sich somit der Bestand dieses Lagers ermitteln. Das virtuelle Lager ist ein wichtiger Bestandteil zur Veranschaulichung des durch Lieferung und Lagerung gebundenen Kapitals.

Diese Sollfortschrittszahlen stellen eine Richtlinie für die Planung der tatsächlichen Produktion dar; sie sind eine Art zeitliche Obergrenze, an der sich orientiert wird, um eine zu späte Auslieferung an den Kunden unbedingt zu vermeiden. Bei der tatsächlich stattfindenden Produktion aber kann es z.B. aus Kostengründen oder aufgrund von vertraglichen Bindungen nötig oder sinnvoll sein, mehrere Produktionen zu einer zusammenzufassen oder auch die Produktion früher als nötig stattfinden zu lassen, um einen gewissen zeitlichen Sicherheitspuffer zu haben. Als Beispiel sei hier ein Kunde genannt, der wöchentlich 500 Exemplare eines Produkts nachfragt. Für das herstellende Unternehmen kann es z.B. aufgrund der Maschinenbeschaffenheit sinnvoll sein, alle zwei Wochen 1000 Exemplare herzustellen und zusätzlich zur Wahrung eines zeitlichen Puffers diese Produktion zwei Tage früher als nötig stattfinden zu lassen. Denkbar wäre auch, am Schluss mehrere Bedarfsmengen zusammenzufassen, um den Kunden aus dem Lager zu beliefern während man die Produktion bereits auf ein neues Produkt umstellt. Den akkumulierten Verlauf der Fortschrittszahlen stellt Abbildung 5 dar.

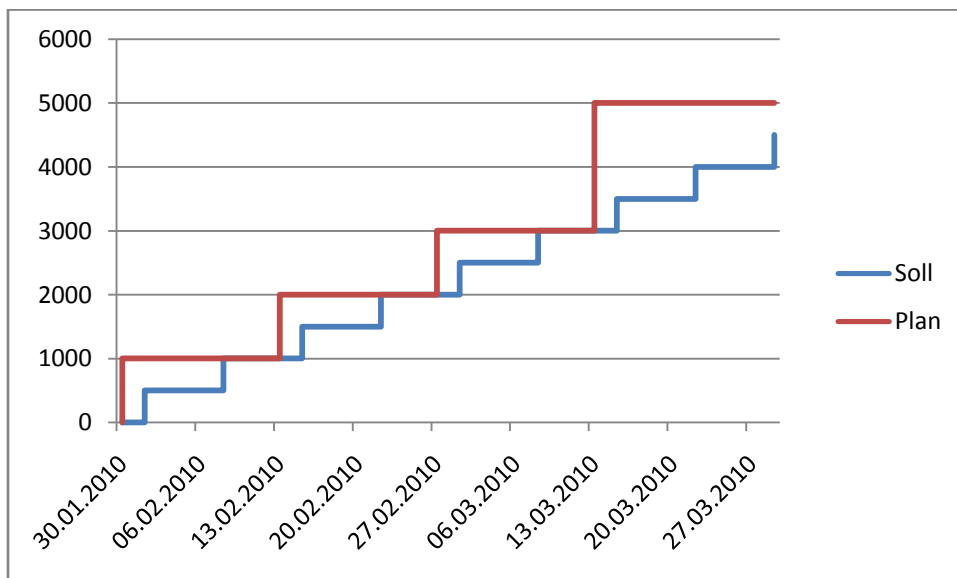


Abbildung 5: Akkumulierter Verlauf der Fortschrittszahlen des Beispiels

Das Ergebnis der Berechnung dieser auch *Produktionslosgrößen* genannten Mengen-Zeitpunkt-Kombinationen sind dann die *Planfortschrittszahlen*. Sie stellen die Zeitpunkte der geplanten Produktion dar und sollten die Sollfortschrittszahlen nicht unterschreiten. Nach der Produktion lassen sich rückblickend die *Ist-Fortschrittszahlen* erstellen, die im Vergleich mit den Planfortschrittszahlen anzeigen, an welchen Stellen Verzögerungen aufgetreten sind. Dadurch ist es dem Unternehmen eventuell möglich, die weitere Produktionsplanung noch einmal anzupassen, um Lieferverzug zu verhindern.

5 Domänenerkundung und Anforderungserhebung

5.1 Chronologischer Ablauf

Das Arbeiten in einer fremden Domäne macht ein gewisses Maß an Grundwissen in diesem Bereich erforderlich, um gewisse Abläufe und Arbeitsweisen zu verstehen. Die Erlangung dieses benötigten Wissens wurde hier in einem ersten Treffen durch persönliche Erläuterungen seitens der zuständigen IPH-Mitarbeiter sowie durch zwei vom IPH zur Verfügung gestellten Artikel ([PlanSt], [TermPA]) zum Thema erreicht. Man bekam so leicht verständlich einen Überblick über produktionslogistische Themen wie Produktausläufe, Lieferketten etc. vermittelt. Weil außerdem in diesem Fall bereits die in Kapitel 3 benannte Excel-Tabelle existierte, die schon einige der Kernfunktionen des zu entwickelnden Programms enthielt, war es angebracht, sich im Anschluss daran in einem weiteren Treffen zunächst einmal die Funktionsweise dieser Tabelle näherbringen zu lassen. So kristallisierten sich bereits die ersten Anforderungen an das Tool heraus und man bekam eine grobe Vorstellung von dem Funktionsumfang, den das Programm am Ende haben sollte.

Da das IPH selbst mit seiner Forschung zu diesem Zeitpunkt noch nicht so weit war, um alle benötigten Funktionen benennen zu können, bestand der nächste Schritt aus dem schriftlichen Auflisten der bisher vereinbarten Anforderungen an das Tool aus dem eigenen Verständnis und der anschließenden Entwicklung eines ersten Prototyps, welcher dann vor allem dem IPH als Anschauungsobjekt diente. Zusammen mit der Anforderungsliste ermöglichte er die Aufdeckung von Missverständnissen und Problemen bei der Umsetzung und half, die Anforderungen zu präzisieren und deren Erhebung voranzutreiben. Damit war es nun möglich, den Prototyp schrittweise weiterzuentwickeln und ihn den Wünschen des IPH anzupassen.

Eine Schwierigkeit zeichnete sich bereits jetzt im Hinblick auf eine – wie man dem Titel der Arbeit entnehmen kann – Kernfunktion der Software ab. Die Algorithmen für die Berechnung des kostengünstigsten Produktauslauftermins und der optimalen Produktionslosgrößen sollten vom IPH geliefert werden, da für ihre Erstellung tiefergehendes produktionslogistisches Fachwissen vonnöten ist. Für die Fertigstellung wurde im Vorfeld zwar ein Termin genannt, jedoch war dieser sehr wagen und auch die Art der Umsetzung und inwieweit sie sich in das Software-Tool integrieren lassen würden waren nicht bekannt. Um das mit diesem Unsicherheitsfaktor verbundene Risiko für die Arbeit abzufedern, wurde in Absprache mit dem IPH eine Methode implementiert, die es ermöglichte, Produktionspläne manuell über Excel-Tabellen einzulesen. Das war zwar nicht so komfortabel, aber es war eine Übergangslösung, die das Weiterarbeiten trotz der später tatsächlich eingetretenen erheblichen Verspätung der Algorithmen ermöglichte.

Von nun an fanden regelmäßige Treffen und reger E-Mail- bzw. Telefonkontakt statt, wodurch der Prototyp immer wieder an die – in diesem speziellen Fall recht hochfrequent anfallenden – Änderungs- und Erweiterungswünsche seitens des IPH angepasst werden konnte. Die wiederholten persönlichen Treffen waren für beide Seiten nötig, um das Verständnis für Probleme und Anliegen des jeweils anderen, fremden Fachbereichs zu erhöhen. Per E-Mail

oder Telefon gab es hierbei mehrfach Verständnisschwierigkeiten, die einfacher in persönlichen Gesprächen ausgeräumt werden konnten.

Ein drittes Treffen diente daraufhin der Besprechung und Überarbeitung der bisher nur rudimentär erhobenen Anforderungen. Das IPH war zu diesem Zeitpunkt inzwischen besser in der Lage, diese Anforderungen zu formulieren, und die anfangs erstellte Anforderungsliste und der bereits weiterentwickelte Prototyp dienten hier als nützliche Grundlage für die Überarbeitungen. So wurde z.B. auch klar, dass anfangs gewünschte Bestandsgraphen, die auf den Sollfortschrittszahlen beruhten und den Lagerbestand der jeweiligen Stufe über die Zeit anzeigen sollten, sich bei näherer Betrachtung aufgrund des homogenen Verlaufs zu wenig Aussagekraft besaßen. Sie wurden gestrichen, um die Übersichtlichkeit der Diagramme zu steigern. Lediglich die Bestandsgraphen der Planfortschrittszahlen wurden beibehalten. Außerdem wurden bei dieser Gelegenheit zusätzlich gleich noch fachbezogene Unsicherheiten aus dem Weg geräumt, beispielsweise ist es einem Laien nicht klar, ob sich der Bestand einer bestimmten Stufe mit dem Eintreffen aller benötigten Rohstoffe oder erst mit Abschluss des Herstellungsprozesses und der damit meist verbundenen temporären Einlagerung des betreffenden Erzeugnisses erhöht.

In der Folge fanden schließlich noch ein viertes und ein fünftes Treffen statt, die in erster Linie sogar vom IPH selbst angestrebt wurden, um die teils massiv veränderten Anforderungen kundtun und auf Umsetzbarkeit prüfen zu können. Im Rahmen dessen wurde z.B. das optische Erscheinungsbild des Software-Tools derart geändert, dass der Diagrammbereich von nun an über Registerkarten in mehrere Sichten auf die Daten und Zusammenhänge mit verschiedenen Funktionen und unterschiedlichem Detailreichtum aufgeteilt wurde. Die Notwendigkeit dieser Aufteilung wurde erst im Laufe der Arbeit klar, als man erkannte, dass die benötigten Daten und Grafiken nicht auf einer Seite unterzubringen waren. Desweiteren war zu diesem Zeitpunkt die Produktauslaufberechnung gerade noch rechtzeitig in einem Stadium angelangt, das die Verbindung mit dem Software-Tool möglich machte. Sie war in Form eines eigenständig aufzurufenden Programms realisiert worden, was die Verbindung erfreulicherweise vereinfachte. Eine vollständige Integration in die Software evtl. inklusive einer sprachlichen Übersetzung des Algorithmus wäre zu diesem fortgeschrittenen Zeitpunkt wohl nicht mehr möglich gewesen.

Nach der Abstimmung mit dem für die Algorithmen zuständigen IPH-Mitarbeiter konnte die Anbindung an das Software-Tool schließlich erfolgen. Ein abschließendes sechstes Treffen diente dann letztlich nur noch der abschließenden Vorführung des fertigen Programms.

5.2 Zusammenfassung und Auswertung der Erfahrungen

Natürlich birgt die Entwicklung einer Software für einen anderen Fachbereich gewisse Risiken und Schwierigkeiten. Das Erkunden dieser fremden Domäne hat zunächst Priorität, da ein gewisses Verständnis der Arbeit und Arbeitsweisen dort von herausragender Wichtigkeit ist. Ohne eine zumindest grundlegende Einarbeitung in diesem Gebiet ist eine solche Arbeit dort nicht möglich.

Die erwähnte Excel-Tabelle eignete sich hierbei optimal dazu, nach der theoretischen Einarbeitung auch einen Bezug zur Praxis zu finden. Sie war schließlich von Mitarbeitern des IPH genau für ihre Zwecke entworfen worden und somit offensichtlich die beste Quelle für die Entdeckung der benötigten Kernfunktionen des zu entwickelnden Programms. Grundsätzlich kann es als lohnenswert erachtet werden, sich anzusehen, wie bisher mit dem zu bearbeitenden Problem umgegangen wurde. Das hilft bei der Schaffung einer Grundlage für weitere Gespräche zu Anforderungserhebung. Außerdem kann man auf diese Weise seine Software an die bisherige Lösung anpassen, um die Umstellung in der Bedienung für die späteren Anwender so leicht wie möglich zu machen.

Ein selbstständiges Erarbeiten und Erkunden der bisherigen Verhältnisse ist in einer fremden Domäne meist nicht möglich. Ein Fachneuling kann auch mit dem vorher angeeigneten Grundwissen nicht in der Lage sein, selbstständig einen Überblick über komplexe, ihm fremde Vorgänge zu erlangen. Gespräche mit den Fachleuten und späteren Anwendern der Software sind daher unumgänglich. Aber auch in einer Domäne, die nicht bzw. nicht mehr fremd erscheint, weil man vielleicht zum wiederholten Male hier arbeitet, ist selbstständiges Erarbeiten nicht zielführend. Die dem zukünftigen Benutzer besonders am Herzen liegenden Funktionen werden auf diese Weise nicht deutlich und man droht, am Anwender vorbei zu entwickeln. Die Folge wäre ein erheblicher Mehraufwand durch nachträgliche Änderungen oder im schlimmsten Fall eine Nichtnutzung des Programms aufgrund von unzureichender Funktionalität.

Als hilfreich hat sich auch die anfängliche Dokumentation der herausgearbeiteten Anforderungen erwiesen. In aus dem eigenen Verständnis niedergeschriebenen Form lässt sich für den Auftraggeber leichter abgleichen, inwieweit und ob überhaupt die erhobenen Anforderungen korrekt verstanden wurden. Möglicherweise aufgetretene Missverständnisse lassen sich so eliminieren bevor unnötig Zeit in eine falsche Umsetzung investiert wird. Jedoch sollte die Dokumentation vorab auch nicht zu voluminös ausfallen. Ein der Informatik fremder Kunde ist den in dieser Arbeit gemachten Erfahrungen nach nicht in der Lage, sich anhand theoretischer Ausarbeitungen eine hinreichend präzise Vorstellung von dem angestrebten Endergebnis zu machen. In der Folge stimmt er eventuell Lösungsvorschlägen zu, die ihm in theoretischer Form vielleicht noch plausibel und angemessen erscheinen, von denen er aber erst nach der Realisierung merkt, dass sie nicht seiner Vorstellung entsprechen. Zeitaufwändige Änderungen werden dann nötig und der Abschluss der Arbeit verzögert sich.

Aus diesem Grund wurde ein iteratives Vorgehen in diesem Projekt vorgezogen und bereits nach dem anfänglichen Anforderungsabgleich ein erster Prototyp entworfen. Insbesondere in

diesem speziellen Fall, bei dem das noch in der Forschung steckende IPH selbst anfangs nicht alle Anforderungen nennen konnte, war eine Umsetzung ohnehin nur Stück für Stück möglich. Mit dem Prototyp bekamen die Mitarbeiter einen besseren Eindruck davon, was in welcher Form realisiert werden konnte und waren somit gleich in der Lage, Änderungs- und Erweiterungswünsche zu formulieren. Er half auch beispielsweise bei der sinnvollen Einteilung des Diagrammbereichs, der zunächst einseitig geplant war. Der Prototyp zeigte hier jedoch den Platzbedarf der einzelnen Komponenten auf und als später sogar noch Diagramme hinzugefügt werden sollten wurde deutlich, dass eine mehrseitige Darstellung vorteilhafter war. Eine im Vorhinein zur Planung angefertigte Zeichnung etwa hätte für den Anwender wenig Praxisbezug gehabt und außerdem nichts über die Realisierbarkeit ausgesagt. Eventuell wäre auch hierbei erst nach der Umsetzung aufgefallen, dass den Diagrammen mehr Platz als geplant eingeräumt werden muss, was wiederum unnötige Zeit zur Nachbesserung in Anspruch genommen hätte. Einen Prototyp hingegen kann der Anwender bereits testweise benutzen und inwieweit die gewünschte Funktionalität gegeben ist.

Darüber hinaus wurden mit Hilfe des Prototyps Missverständnisse entdeckt – Dinge, die anders umgesetzt wurden als erwünscht - sowie Probleme und Einschränkungen bei der Umsetzung, an die vorher niemand gedacht hatte, deutlich gemacht. Als Beispiel sei hier die erwünschte Funktion zum Einlesen der Lieferketten genannt, die aus einer mit den entsprechenden Daten gefüllten Excel-Tabelle die benötigte Datenstruktur im Software-Tool erzeugen sollte. Den Mitarbeitern des IPH wurde erst durch den Prototyp bewusst, dass für diese Excel-Tabellen ein festes Format festgelegt werden musste, damit das Einlesen durch eine Software zuverlässig funktioniert. Über diese Funktion war im Vorfeld nicht ausreichend gesprochen worden, sodass das sich testweise implementierte Format als unpraktisch herausstellte, da die Tabellen von den durch die Systeme des IPH erstellten Tabellen stark abwichen. Der Vorfall beweist wieder die Bedeutung ausgiebiger Gespräche im Vorhinein, zeigt aber auch, dass ohne den Prototyp das Problem erst wesentlich später aufgefallen und Beseitigung damit zeitaufwändiger gewesen wäre. Die Festlegung eines passenden Formats konnte so nachgeholt und das Modul zum Einlesen entsprechend angepasst werden.

Eine Einschränkung musste dann bei der Art der Darstellung dieser Lieferkette im Programm hingenommen werden. Gewünscht war zwar eine der Abbildung 3 gleichende Darstellung, jedoch wurde erst im Nachhinein klar, dass die theoretisch nicht begrenzte Größe einer solchen Kette eine solche Darstellung aufgrund des Platzmangels unmöglich macht. Im gegenseitigen Einvernehmen einigte man sich hier auf eine Baumdarstellung, wie man sie vom Windows-Explorer kennt. In einer theoretischen Ausarbeitung hätte man die gewünschte Darstellung vielleicht eingeplant, hätte allerdings bei der Realisierung am Ende festgestellt, dass sie nicht umsetzbar ist. Daraus kann sich dann angesichts der fortgeschrittenen Zeit ein Problem entwickeln, wenn der Anwender sich mit der alternativen Darstellung nicht einverstanden zeigt.

Als mit Abstand größtes Risiko für die Arbeit konnte sicherlich die Abhängigkeit von den in Fremdarbeit programmierten Losgrößenalgorithmen eingestuft werden. Ihre - verglichen mit dem zu Beginn genannten Fertigstellungstermin - enorme Verspätung bestätigte dies im Nachhinein. Eine weitere Woche Verspätung hätte den Zeitrahmen gesprengt und die Imple-

mentierung dieser Kernfunktion verhindert, was durchaus negative Folgen für die Arbeit gehabt hätte. Die Einsetzbarkeit der Software wäre stark eingeschränkt worden und die Frage nach dem Bezug der Software zum Titel der Arbeit wäre berechtigt gewesen. Die eingesetzte Übergangslösung gewährleistete – wenn auch eingeschränkt komfortabel – die Funktion und erwies sich daher in diesem speziellen Fall als probates Mittel, eine Gefährdung der Arbeit auszuschließen. Diese Erfahrung lässt den Schluss zu, dass der Einsatz von Ausweichlösungen zur Entschärfung von unbeeinflussbaren Risikofaktoren generell als nützlich und hilfreich angesehen werden kann. Wann immer man bei seiner Arbeit von dem Erfolg Anderer abhängig ist, sichern sie den eigenen Erfolg bei vergleichsweise geringem zeitlichem Zusatzaufwand ab. Sicherlich wäre es in diesem speziellen Fall, bei dem der Kunde selbst für die Verspätung der einzubindenden Fremdmodule verantwortlich war, eine mögliche Alternative gewesen, das Programm mit einer Schnittstelle zu versehen, an die sich das IPH bei der Programmierung des Produktionslosgrößenalgorithmus hätte halten müssen. Zuvorkommender ist es jedoch, sich an die Schnittstellen des Kunden anzupassen, erst recht wenn dieser womöglich nicht das nötige Informatik-Fachwissen besitzt, in seiner Programmierung die Schnittstelle eines anderen Programms zu nutzen.

Bei diesem Projekt gab es relativ viele Treffen, die der Abstimmung dienten, und abseits dessen auch häufigen Telefon- und E-Mail-Kontakt. Sicherlich wurde dies hauptsächlich nötig durch die Anforderungen, die nur Stück für Stück geäußert werden konnten, aber es erleichterte auch die Einarbeitung in den fremden Fachbereich. Abschließend kann gesagt werden, dass beim Arbeiten für eine fremde Domäne die ausreichende Kommunikation eines der wichtigsten Erfolgskriterien ist, da beide Seiten in dem Bereich der jeweils anderen Seite bestenfalls über Grundwissen verfügen. Nach einem anfänglichen Abstimmungstreffen die komplette Software ohne weitere Zusammenarbeit programmieren zu wollen, wird nicht zu einem zufriedenstellenden Ergebnis führen. Ein reger Kontakt hilft beiden Seiten. Dabei sind persönliche Treffen der Kommunikation über E-Mail oder Telefon immer vorzuziehen; der direkte Kontakt erleichtert den Informationsaustausch.

6 Anforderungen an das Software-Tool

Im Folgenden werden die endgültig umgesetzten Anforderungen zusammenfassend genannt.

1. Die Lieferkette muss über Excel-Tabellen in die Software eingelesen werden können. Diese Listen liegen bereits vor und haben ein festes Format. Wie aus Abbildung 6 ersichtlich wird, stellt jede Zeile dieser Tabelle eine Lieferkettenstufe mit all ihren einzulesenden Parametern dar.

	A	B	C	D	E	F	G	H	I	J
1	Nummer	Bezeichnung	ist Bestandteil von	Gewichtung	Lieferkettenpartner	Sicherheitszeit/Produktionszeit	Transportzeit	Stückkosten	Rüstkosten	Bestellkosten
2	100150	Zusammenbau, Vertrieb		1,00	Gärtner	3	3	300,00	100,00	20,00
3	200100	Kontaktierung	100150	1,00	Richter	5	3	150,00	0,00	20,00
4	300100	Elektronik Teil1	200100	1,00	Pottthast	10	3	80,00	100,00	20,00
5	400100	Elektronik Teil2	200100	1,00	Knigge	8	3	70,00	200,00	20,00
6	300200	Gehäuse	100150	1,00	Dömming	5	3	20,00	100,00	20,00
7	300300	Kunststoff	300200	5,00	Prüssing	2	3	3,00	0,00	20,00

Abbildung 6: Darstellung einer Lieferkette in Tabellenform

2. Diese Lieferkette soll anschließend in einer als Baum zu verstehenden Form im Programm dargestellt werden. Die Stufen müssen einzeln auswählbar sein, um die passenden stufenspezifischen Diagramme anzeigen zu können.
3. Die in einem Informationspanel unterhalb der Lieferkette angezeigten Parameter der ausgewählten Stufe (Produktionsdauer, Gewichtung, Stückkosten etc.) sollen aus Gründen der Übersichtlichkeit keine Nullwerte enthalten. Parameter, für die der Wert „0“ angegeben wurde, sollen ausgeblendet werden.
4. Der Kundenbedarf liegt als eine der Abbildung 7 ähnlichen Liste von Zeitpunkten, der zu diesen Zeitpunkten vom Kunden bestellten Warenmenge und gegebenen-

	A	B	C	D
1	Datum	Kundenbedarf [Stück]	reduzierter Kundenbedarf	erhöhter Kundenbedarf
2	25.04.2010	5.000,00	5.000,00	5.000,00
3	26.04.2010	4.500,00	4.300,00	4.600,00
4	27.04.2010	4.300,00	4.034,00	4.433,00
5	28.04.2010	4.200,00	3.867,00	4.367,00
6			

Abbildung 7: Beispiel einer Kundenbedarfstabelle

falls einer mengenmäßigen Abweichung nach oben und unten vor. Diese Listen sind ebenfalls im Excel-Format und müssen von dem Software-Tool eingelesen und verarbeitet werden können.

5. Nach dem Einlesen soll der Kundenbedarf grafisch als Verlauf der Menge über die Zeit dargestellt werden. Die Graphen für die Abweichungen sollen im gleichen Diagramm Platz finden. Ist keine Abweichung angegeben, sollen diese Graphen natürlich nicht zu sehen sein.
6. Für den Diagrammbereich wird eine Aufteilung in unterschiedliche Sichten gewünscht.
 - a. Sicht „Kundenbedarf“

In dieser Sicht soll der Kundenbedarf wie in Punkt 5 beschrieben angezeigt werden.
 - b. Sicht „Lieferkette – Übersicht“

Diese Sicht dient der Übersicht über die Lieferkette an der ausgewählten Stelle. Das Fortschrittszahlendiagramm der ausgewählten Lieferkettenstufe ist hierbei mittig anzuzeigen. Das des Nachfolgers (falls vorhanden) soll sich rechts davon befinden, die Diagramme von bis zu drei Vorgängern links. Diese Diagramme enthalten allein die Sollfortschrittszahlen.
 - c. Sicht „Lieferkette“

Die Visualisierung der Zulieferbeziehungen der ausgewählten Lieferkettenstufe ist die Aufgabe dieser Sicht. Dabei sollen sich zwischen den Sollfortschrittszahlendiagrammen der ausgewählten Stufe und ihren bis zu drei Vorgängern die virtuellen Lager der Vorgänger befinden.
 - d. Sicht „Detailansicht“

Diese detailliertere Sicht enthält ein großes Fortschrittszahlendiagramm mit Soll- und Planfortschrittszahlen sowie noch einmal das Diagramm des virtuellen Lagers für die ausgewählte Stufe, das nun auch einen Graphen für den Lagerbestand enthält.
7. In allen Diagrammen soll der aktuelle Tag durch eine senkrecht verlaufende Linie kenntlich gemacht werden. Außerdem wurde gefordert, die x-Achse mit dem Datum und die y-Achse mit der Menge zu beschriften. Die Fortschrittszahlen werden stets akkumuliert dargestellt, um die Aussagekraft zu steigern.
8. Es wurde gefordert, in den Diagrammen die Darstellung auf ausgewählte Zeit- und Mengenbereiche einschränken zu können. Diese Möglichkeit dazu ist allerdings in Form einer Zoomfunktion in die für die Erstellung der Diagramme verwendete Bibliothek bereits implementiert und bedarf keiner weiteren Schritte.

9. Die Verwendung des externen Losgrößenberechnungsprogramms zur Berechnung der Planfortschrittszahlen erfordert es außerdem, die Sollfortschrittszahlen in eine dem Programm beiliegende Datenbank zu schreiben. Der anschließende Aufruf des Programms stößt die Berechnungen an. Nach Fertigstellung sind die Ergebnisse ebenfalls in der Datenbank verfügbar und können ausgelesen werden.
10. Die Übergangslösung, die das manuelle Einlesen der Produktionslosgrößen über Excel-Tabellen ermöglichte, wurde in der Software belassen. Somit ist es immer noch möglich, alternativ dieses manuelle Einlesen zu verwenden. Es werden hierbei Excel-Tabellen mit dem gleichen Format wie die Kundenbedarfstabelle verwendet.

In erster Linie wird das Software-Tool von den Mitarbeitern des IPH in der produktionslogistischen Forschung zur Visualisierung der Auswirkungen unterschiedlicher Lieferketten- und Parameterkonfigurationen verwendet. Die Benutzer werden also im Allgemeinen mit der Materie vertraut sein und bedürfen damit keiner besonderen Erklärung oder Einarbeitung. Da die vom IPH entwickelte Excel-Tabelle die Grundlage des Software-Tools ist, ist es für diese Anwendergruppe weitestgehend intuitiv zu bedienen.

7 Der ALEKS Demonstrator: Aufbau und Struktur

Das im Rahmen dieser Arbeit entwickelte Software-Tool – im Folgenden gemäß seiner in erster Linie visualisierenden Funktion auch „Demonstrator“ genannt – erhielt den Namen „ALEKS“ und wurde damit nach dem laufenden Forschungsprojekt des IPH (Auslaufplanung in Lieferketten der Elektronikindustrie) benannt. Der innere Aufbau des Programms und das Gefüge der einzelnen Komponenten sollen hier erklärt und mit Hilfe einiger UML-Diagramme veranschaulicht werden. Diese Diagramme halfen bei der Planung und während der Erstellung des Tools bei der Gewinnung der nötigen Übersicht über die Struktur. Sie ermöglichen darüber hinaus unter der Voraussetzung der nötigen UML-Kenntnisse die Einarbeitung Anderer in den Programmcode und vereinfachen so die Implementierung möglicher zukünftiger Erweiterungen.

7.1 Das Produkt und seine Lieferkette

Das Herzstück des Demonstrators ist die im Package „SupplyChain“ untergebrachte Datenstruktur zur Speicherung und Verwaltung der Lieferkette, ihrer einzelnen Stufen und des Kundenbedarfs (Abbildung 8).

Das zentrale Element ist hier die Klasse „Product“. Sie bekommt einen passenden Namen und symbolisiert das jeweilige Endprodukt und speichert die zugehörige Lieferkette in Form eines Verweises auf die Wurzel. Außerdem implementiert sie das Java-interne „TreeModel“, das die Baumdarstellung als „JTree“ ermöglicht. Diese in Java bereits implementierte Klasse automatisiert Darstellungen von Baumstrukturen und ermöglicht Zugriffe des Benutzers auf jeden einzelnen Knoten.

Wie schon die Wurzel selbst, sind auch die einzelnen Stufen der Kette Objekte der Klasse „Supplier“, also Zulieferer. Die verkettete Struktur wird dadurch erreicht, dass jeder Zulieferer Verweise auf seinen Nachfolger und Vorgänger enthält. Eine Stufe kann dabei mehrere Vorgänger, jedoch nur einen Nachfolger besitzen. Neben der Möglichkeit, dem von diesem Zulieferer hergestellten Zwischenprodukt einen Namen und eine Artikelnummer zu geben, werden hier in weiteren Variablen auch die ganzen für Berechnungen benötigten Parameter bereit gehalten.

Der Kundenbedarf ist eine Liste von Objekten der Klasse „Order“, einer Kombination aus Datum und Menge, und wird in seiner Urform im Produkt gespeichert. Die bei der Berechnung der Fortschrittszahlen entstehenden Listen, die den um die jeweiligen Produktions- und Lieferzeiten verschobenen Kundenbedarf enthalten, werden in den Zulieferern gespeichert. Die Interfaces der Klassen „Supplier“ und „Order“ wurden implementiert, um später auch alternative Realisierungen dieser Klassen zu ermöglichen.

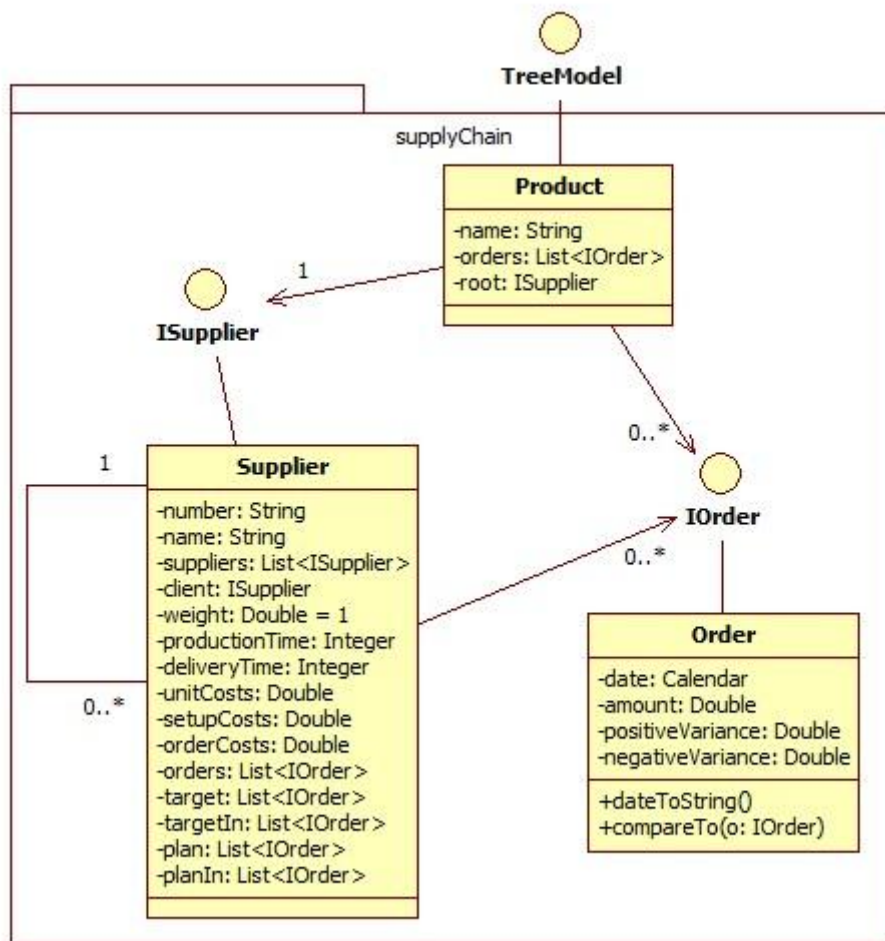


Abbildung 8: Das Package "SupplyChain"

7.2 Dateioperationen

Das Package „FileOperations“ (Abbildung 9) liefert die für die Dateioperationen benötigten Klassen.

Die Klasse „FileChooser“ dient dabei der Dateiauswahl. Sie öffnet den Datei-öffnen-Dialog, sobald der Anwender das Einlesen einer Datei wünscht, und liefert die ausgewählte Datei zurück. Sie ist eine Erweiterung der Java-internen Klasse „JFileChooser“ und ermöglicht, die Dateiauswahl auf einige voreingestellte Dateitypen über den Parameter „mode“ einzuschränken. Derzeit wird die Klasse immer mit Einschränkung auf Excel-Dateien (*.xls) verwendet, da das Einlesen anderer Dateitypen noch nicht vonnöten ist.

Die ausgewählten Dateien werden je nach Verwendungszweck entweder der Klasse „XlsSupplyChainReader“ zum Einlesen der Lieferkette oder der Klasse „XlsOrdersReader“ zum Einlesen der Kundenbedarfsliste übergeben. In beiden Klassen ist in „firstLine“ die Zeile festgelegt, in der die Daten in der Excel-Datei beginnen müssen, damit Anpassungen bei Änderung des Tabellenformats möglich sind. Außerdem speichern sie beim Einlesevorgang auf-

getretene Fehler in Form einer Liste der betroffenen Zeilen. Das ermöglicht hinterher die Ausgabe in einer Fehlermeldung. Der jeweilige Rückgabewert ist dann je nach Klasse entweder die Lieferkette repräsentiert durch ihre Wurzel oder aber die Kundenbedarfsliste. Auch hier sind Interfaces verwendet worden, um beispielsweise auch ein Einlesen aus anderen Dateiformaten einbinden zu können.

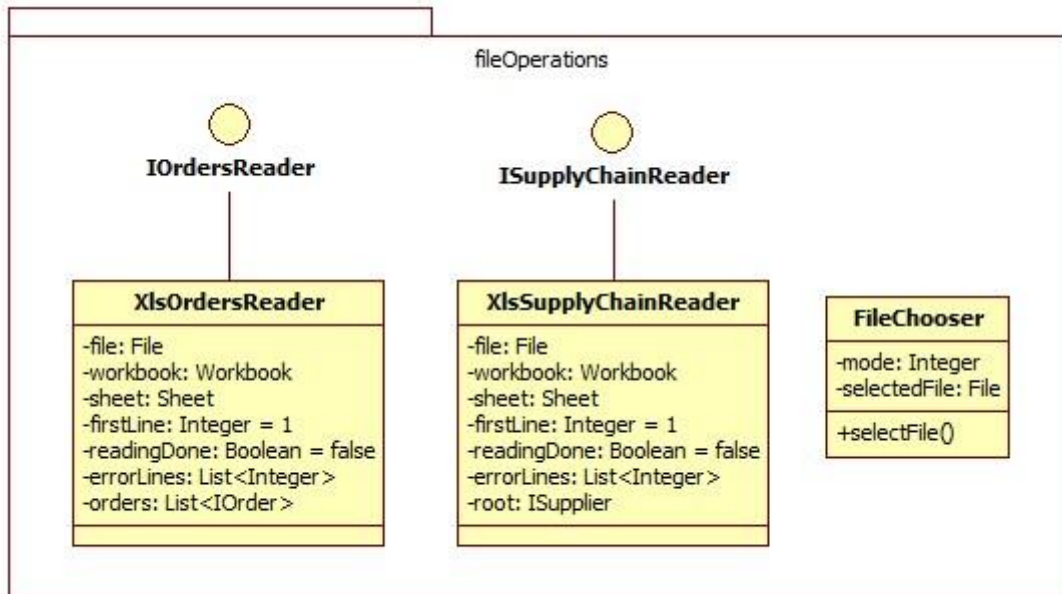


Abbildung 9: Das Package "FileOperations"

7.3 Diagramme

Das Package „Graphics“ ist für alle Diagramme des Demonstrators zuständig. Wie Abbildung 10 zeigt, sind dafür nur zwei Klassen nötig, da sie flexibel genug programmiert wurden, um alle benötigten Ansichten der Diagramme liefern zu können.

Beide erben von der Klasse „JPanel“, damit sie einfach in die Swing-Oberfläche des Software-Tools integriert werden können. Alle Graphen, die hier angezeigt werden können, werden hier aus den Listen der dem Diagramm zugewiesenen Lieferkettenstufe erzeugt und gespeichert. Zusätzlich erzeugen sie die Markierung für den aktuellen Tag. Über boolesche Variablen ist es darüber hinaus möglich, die Anzeige einzelner Graphen und der Legende ein- und auszuschalten. Das ist nötig, weil einige Diagramme Planfortschrittszahlen oder Bestände enthalten, andere aber nur Sollfortschrittszahlen anzeigen sollen und Legenden nicht bei jedem Diagramm wiederholt werden müssen. Objekten der Klasse „SupplierChartPanel“ ist es darüber hinaus möglich, durch das Setzen des Parameters „warehouse“ ein Diagramm für das nachgelagerte virtuelle Lager des Zulieferers zu erstellen.

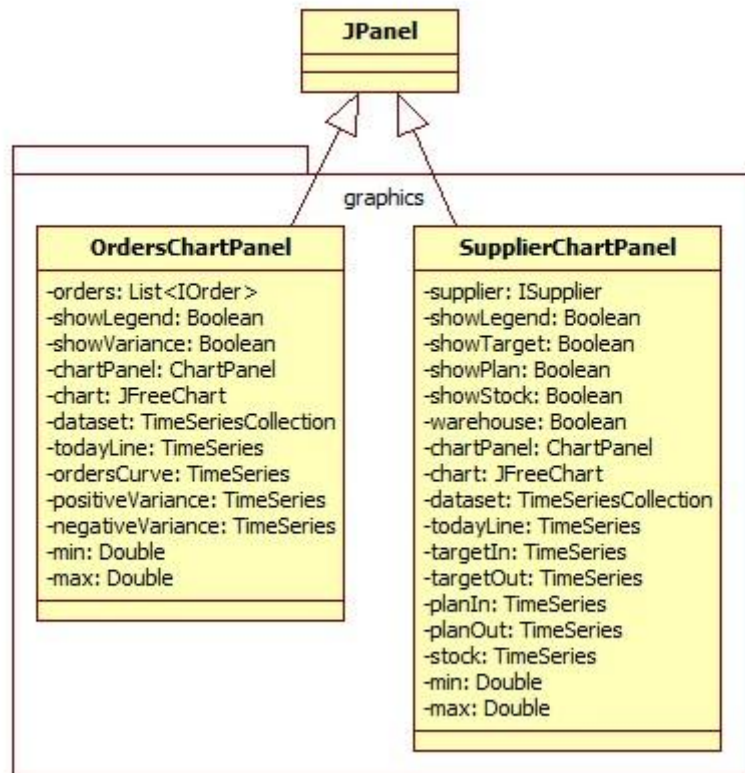


Abbildung 10: Das Package "Graphics"

7.4 Der Losgrößenalgorithmus

Das Package „Algorithms“ (Abbildung 11) ermöglicht den Zugriff auf das externe Programm zur Berechnung der Produktionslosgrößen und Produktausläufe, das den sogenannten Wagner-Within-Algorithmus verwendet. Die einzige enthaltene Klasse ist nach diesem Algorithmus benannt und sorgt als zentrale Schnittstelle für die Kommunikation mit der Datenbank und den Aufruf des externen Programms. Die Datenbank muss dabei als ODBC-Datenquelle eingerichtet sein; der Name dieser Quelle lässt sich in der Variable „connectionString“ angeben. Die Klasse schreibt schließlich die Sollfortschrittszahlen des betreffenden Zulieferers in die Datenbank und startet das externe Programm. Nach Fertigstellung kann sie die Ergebnisdaten als Planfortschrittszahlen an den Zulieferer zurückliefern. In diesem Package lassen sich zur Verwendung alternativer Algorithmen natürlich noch weitere Klassen unterbringen.

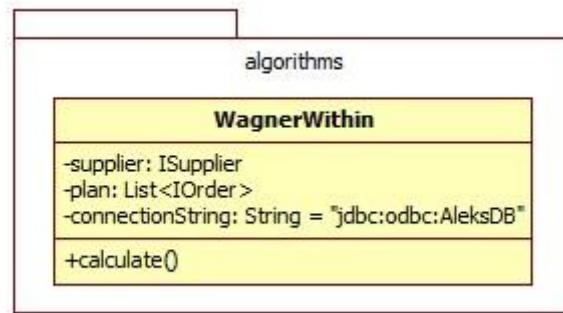


Abbildung 11: Das Package "Algorithms"

7.5 Die grafische Benutzerschnittstelle (GUI)

Für die Verbindung der einzelnen Komponenten sorgt das Package „GUI“, dessen Struktur Abbildung 12 illustriert. Die einzelnen Klassen dienen der Erzeugung der grafischen Oberfläche, welche die Bedienung durch den Anwender ermöglicht. Die zentrale Klasse „MainWindow“ speichert das hier verwendete Produkt und erzeugt dazu das Hauptfenster des Demonstrators. Dieses Fenster enthält einige wichtige Elemente wie etwa das JTree-Element zur Baumdarstellung der Lieferkette oder die Schaltflächen zum Einlesen der Dateien; die restlichen Elemente sind in die anderen Klassen des Packages ausgelagert. Dazu gehört auch das Informationspanel, das die Parameter der ausgewählten Lieferkettenstufe anzeigt. Ebenso wird jede der geforderten Sichten durch eine eigene Klasse erzeugt. So lassen sich die Klassen unabhängig voneinander verändern und weitere Sichten einfach hinzufügen.

Die Klasse „OrdersView“ sorgt dabei für die Darstellung des Kundenbedarfs, während „SupplyChainOverview“ und „SupplyChainView“ die beiden unterschiedlich detaillierten Sichten auf die Lieferkette erzeugen. Für die Erstellung der Diagramme werden die Klassen aus dem Package „Graphics“ genutzt. Die Klasse „DetailView“ für die Detailansicht enthält neben den benötigten Diagrammen zusätzlich die Schaltflächen zum Start der Berechnung durch den Algorithmus und zum manuellen Einlesen und Löschen des Produktionsplans.

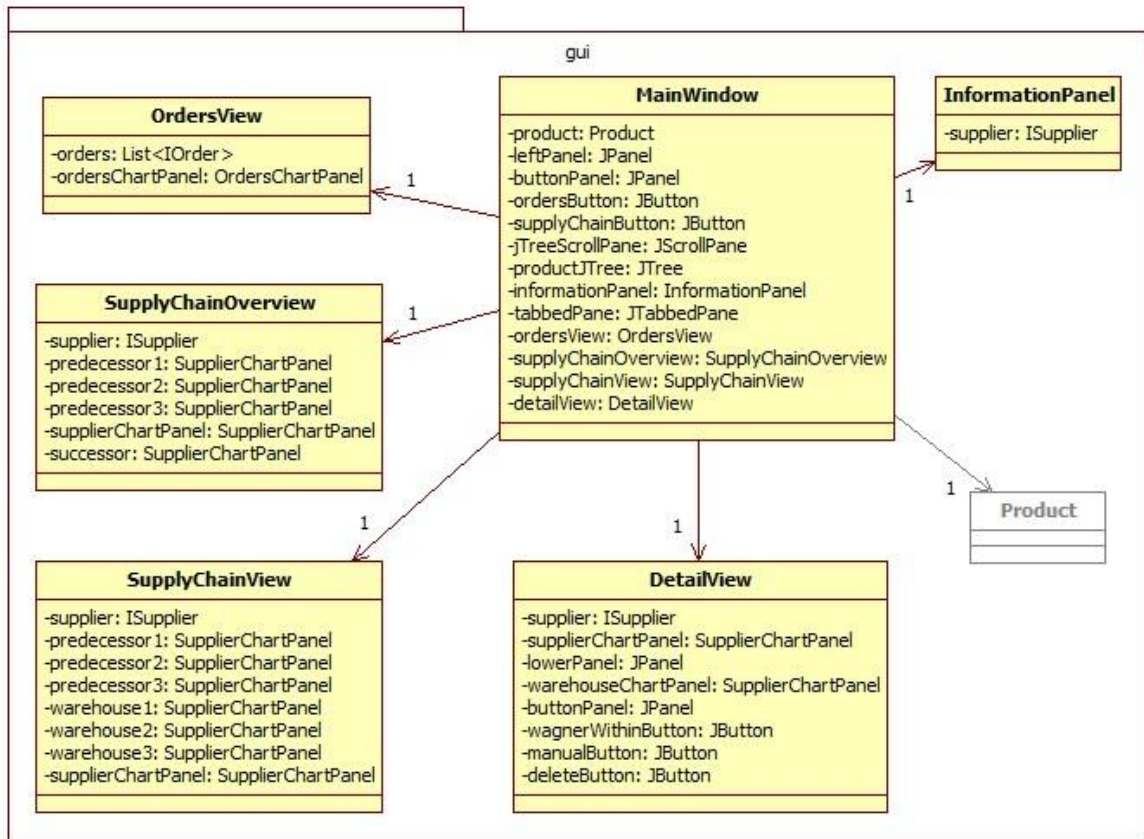


Abbildung 12: Das Package "GUI"

8 Ausblick und kritische Würdigung

Beim ALEKS Demonstrator wurde dafür gesorgt, dass er an den wichtigen Stellen erweiterbar ist. Wie das vorangegangene Kapitel gezeigt hat sind Klassen, die wichtige Funktionen realisieren, mit Interfaces versehen, sodass Erweiterungen oder Veränderungen an diesen Stellen einfach fallen, ohne große strukturelle Umbauten mit sich zu ziehen. Die Trennung der Quelldaten in Excel-Dokumente einerseits und die für die Losgrößenberechnung benötigte Datenbank andererseits ist beispielsweise eine momentan noch etwas unglückliche Lösung. Eine sinnvolle Alternative wäre es hier, mit den Lieferketten, dem Kundenbedarf und den weiteren Parametern alle Daten der Produkte komplett innerhalb einer Datenbank zu verwalten. Da das aber eine erhebliche Umgestaltung bei der Organisation der Quelldaten mit sich zöge, läge der Hauptanteil der Arbeit an dieser Änderung beim IPH. Die nötige Änderung am Demonstrator hingegen fällt aufgrund der Struktur der Software weniger ins Gewicht.

Außerdem fehlt dem Demonstrator bisher eine Funktion zu Verwaltung der Ist-Fortschrittszahlen. Damit wäre es dem Anwender möglich die Soll- und Planfortschrittszahlen, die vorher einen Blick auf die Zukunft gewährten, nach der Produktion mit den tatsächlich eingetretenen Verhältnissen zu vergleichen. Unterschiede können die Folge einer Fehlplanung sein, die der Benutzer so aufdecken und in Zukunft vermeiden könnte. Eine mögliche Ursache könnte auch ein chronisch auftretender Engpass innerhalb der Lieferkette sein, der regelmäßig für Verzögerungen sorgt. Dessen Entdeckung würde die Vergrößerung eines zeitlichen Sicherheitspuffers ermöglichen.

Eine weitere Modifikation wäre möglich bei der Bestimmung der Abweichungen des Kundenbedarfs (Abbildung 13). Derzeit muss innerhalb der einzulesenden Excel-Tabelle für jede Menge einzeln die Abweichung nach oben und nach unten angegeben werden. Hier wäre eine Automatisierung des Vorgangs möglich, sodass der Anwender lediglich Angaben zu Beginn- und Endzeitpunkt sowie zur maximalen Höhe der Abweichung machen muss. Der Graph wird dann durch das Software-Tool berechnet. Eine solche Änderung war bereits im Gespräch, hier wurde jedoch der Realisierung anderer Funktionen mit höherer Priorität Vorrang gewährt.

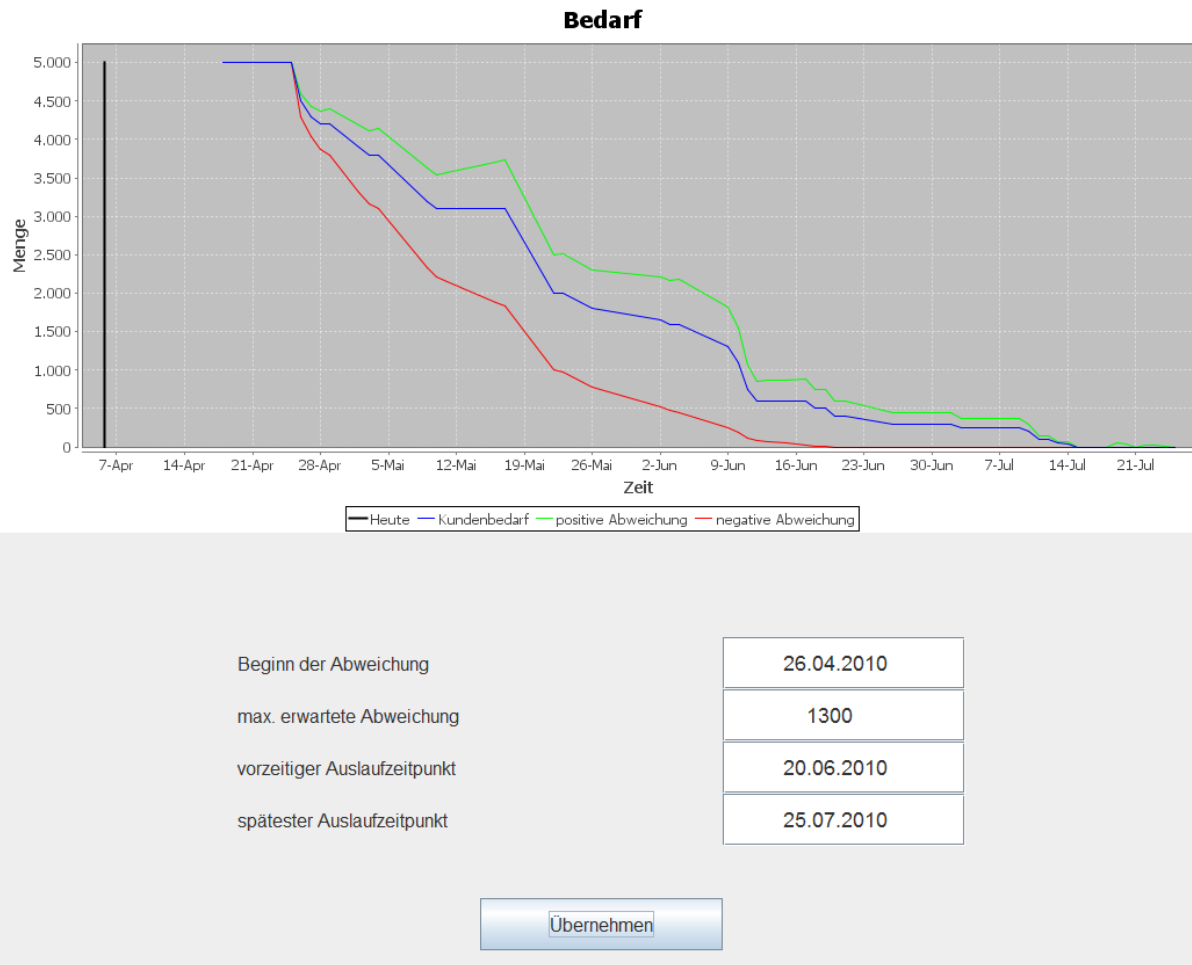


Abbildung 13: Eine Möglichkeit zur Steuerung der Abweichungsberechnung durch den Benutzer

Wahrscheinlich wäre es nützlich, dem Demonstrator darüber hinaus eine Speicherfunktion zu verleihen. Der Benutzer hätte dann die Möglichkeit, für jedes betrachtete Produkt eine Speicherdatei anzulegen, mit der er den momentanen Zustand des Demonstrators jederzeit wiederherstellen kann, ohne Daten erneut einlesen oder Eingaben wiederholen zu müssen. Das würde den Arbeitsaufwand vor allem bei einem häufigen Wechsel des angezeigten Produkts vermindern. Im Zusammenhang mit der als erstes in diesem Kapitel vorgestellten Änderung könnte die Datenbank sogar dahingehend erweitert werden, dass sie die Speicherdateien mit den jeweiligen in ihr verwalteten Produkten verknüpft.

9 Literaturverzeichnis

[TermPA] Hertrampf, Frauke und Prof. Dr.-Ing. habil. Nyhuis, Peter. 2009.

Terminierung von Produktausläufen in Lieferketten in der Elektronikindustrie. Hannover, Deutschland : s.n., 2009.

[PlanSt] Hertrampf, Frauke, Dr.-Ing. Dipl.-Oec. Nickel, Rouven und Prof. Dr.-Ing. habil. Nyhuis, Peter. 2009. Planung und Steuerung von Produkt- und Baugruppenausläufen in Lieferketten der Elektronikindustrie. Hannover, Deutschland : s.n., 2009.

[Hom09] Homburg, Christian und Krohmer, Harley. 2006. *Grundlagen des Marketingmanagements*. Mannheim, Bern : Gabler, 2006.

10 Abbildungs- und Tabellenverzeichnis

Abbildung 1: Ausschnitt aus der bisher verwendeten Excel-Tabelle (Zoomfaktor 10%).....	6
Abbildung 2: Absatzverlauf im Produktlebenszyklus	8
Abbildung 3: Beispiel einer Lieferkette	9
Abbildung 4: Berechnung der Sollfortschrittszahlen in einer Lieferkette	11
Abbildung 5: Akkumulierter Verlauf der Fortschrittszahlen des Beispiels.....	12
Abbildung 6: Darstellung einer Lieferkette in Tabellenform.....	18
Abbildung 7: Beispiel einer Kundenbedarfstabelle.....	18
Abbildung 8: Das Package "SupplyChain"	22
Abbildung 9: Das Package "FileOperations"	23
Abbildung 10: Das Package "Graphics"	24
Abbildung 11: Das Package "Algorithms"	25
Abbildung 12: Das Package "GUI"	26
Abbildung 13: Eine Möglichkeit zur Steuerung der Abweichungsberechnung durch den Benutzer	28
Tabelle 1: Parameter einer Lieferkettenstufe	10

11 Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 12.04.2010

Lars Wolter