

Gruppe P0044

IBSYS-II

Produktions – und Planungstool



Marc Riar

Dennis Söhnen

Azizbek Atabekov

Inhalt

1. Ausgangssituation	2
2. Aufgaben des Produktions- und Planungstools	2
2.1 Disposition Eigenfertigung	3
2.2 Kaufteildisposition.....	3
2.3 Kapazitätsplan	3
3. Einbettung des PPT in das Gesamtsystem	3
4. Bedienungsanleitung.....	5
5. Umsetzung.....	10
5.1 Eingesetzte Technologien.....	10
5.2 Implementierung.....	10
5.2.1 Implementierung Disposition Eigenfertigung	10
5.2.2 Implementierung Kaufteildisposition.....	10
5.2.3 Implementierung Kapazitätsplan	12

1. Ausgangssituation

Im Rahmen der Veranstaltung IbSys II wurde ein Planspiel durchgeführt bei dem Vertriebsentscheidungen getroffen werden mussten und verschiedene Logistik-Aufgaben zu bearbeiten waren. Im Groben, gehörten zu den Logistik-Aufgaben die Ermittlung der Produktionsaufträge für die jeweils kommende Periode, eine Kapazitätsplanung mit Festlegen von Schichten und Überstunden sowie die Durchführung des strategischen und operativen Einkaufs. Grundlage war dabei der Supply Chain Simulator (www.scsim.de). Es handelt sich hierbei um ein Trainingstool zur Simulation von Logistikketten.

Um die oben genannten Logistik-Aufgaben schnell und zuverlässig durchführen zu können wurde ein Produktions- und Planungstool entwickelt, welches die benötigten Kennzahlen automatisch berechnet. Dieses Produktions- und Planungstool wird im folgenden Handbuch vorgestellt.

2. Aufgaben des Produktions- und Planungstools

Das PPT hat die Aufgabe, dem Benutzer die Bewältigung der im grünen Kasten aufgeführten Logistik-Aufgaben abzunehmen.



2.1 Disposition Eigenfertigung

Das PPT soll unter Berücksichtigung von Vertriebswunsch, dem geplanten Lagerbestand am Ende der Planperiode, dem Lagerbestand am Ende der Vorperiode, den Aufträgen in der Warteschlange und den Aufträgen in Bearbeitung, die Produktionsaufträge für die kommende Periode berechnen.

2.2 Kaufteildisposition

Bei der Aufgabe der Kaufteildisposition soll das PPT entscheiden ob ein bestimmtes Kaufteil bestellt werden soll oder nicht, bzw. ob eine Normal- oder Eilbestellung vorliegt. Um diese Entscheidung treffen zu können, müssen die Forecasts, der aktuelle Vertriebswunsch, der aktuelle Lagerbestand sowie die Lieferfrist und Abweichung berücksichtigt werden.

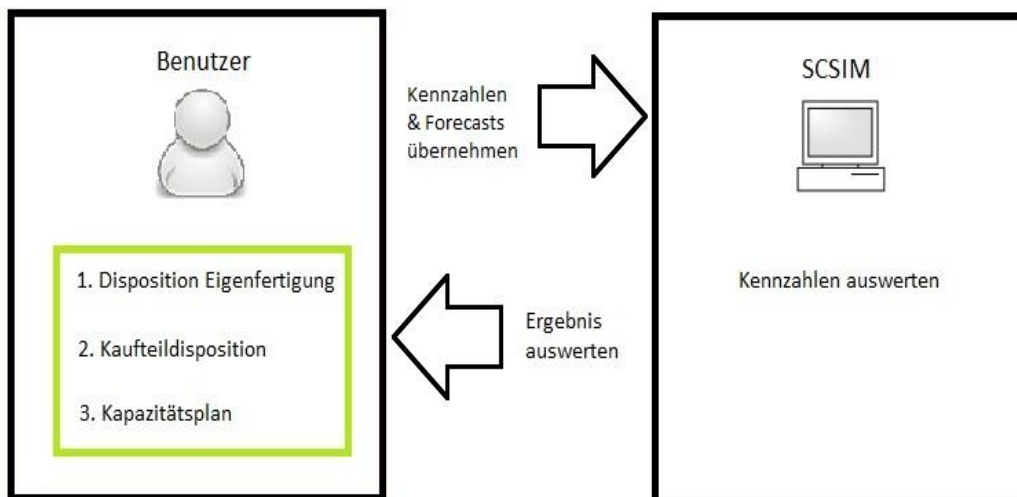
2.3 Kapazitätsplan

Das PPT soll ermitteln, wie viele Schichten und Überstunden in der kommenden Periode geleistet werden müssen. Dabei wird für jeden der vierzehn Arbeitsplätze der Gesamtkapazitätsbedarf ermittelt.

3. Einbettung des PPT in das Gesamtsystem

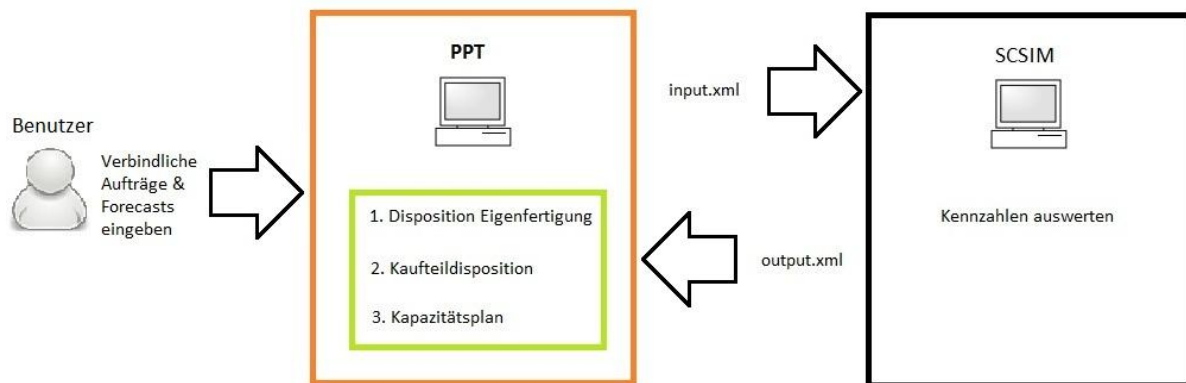
Bevor das Produktions- und Planungstool (PPT) im Einsatz war mussten die Logistik-Aufgaben Disposition Eigenfertigung, Kaufteildisposition, Kapazitätsplan vom Benutzer selbst ermittelt werden.

Die folgende Grafik beschreibt den Sachverhalt ohne PPT.



Nachdem der Benutzer die im grünen Kasten aufgeführten Logistik-Aufgaben per Hand oder mit Hilfsmitteln (z.B. Excel) bewältigt hatte, mussten die Kennzahlen und Forecasts in den Supply Chain Simulator (SCSIM) übertragen werden. Der SCSIM errechnete die Werte für die aktuelle Periode mittels der Kennzahlen. Anschließend wurde das Ergebnis vom Benutzer überprüft und ausgewertet. Anhand dieser Auswertung wurden Entscheidungen für die kommende Periode getroffen und schließlich erneut die Rechnungen vom Benutzer durchgeführt. Bevor das PPT eingesetzt wurde war dies ein sehr zeitaufwendiger Prozess.

Sachverhalt mit PPT.



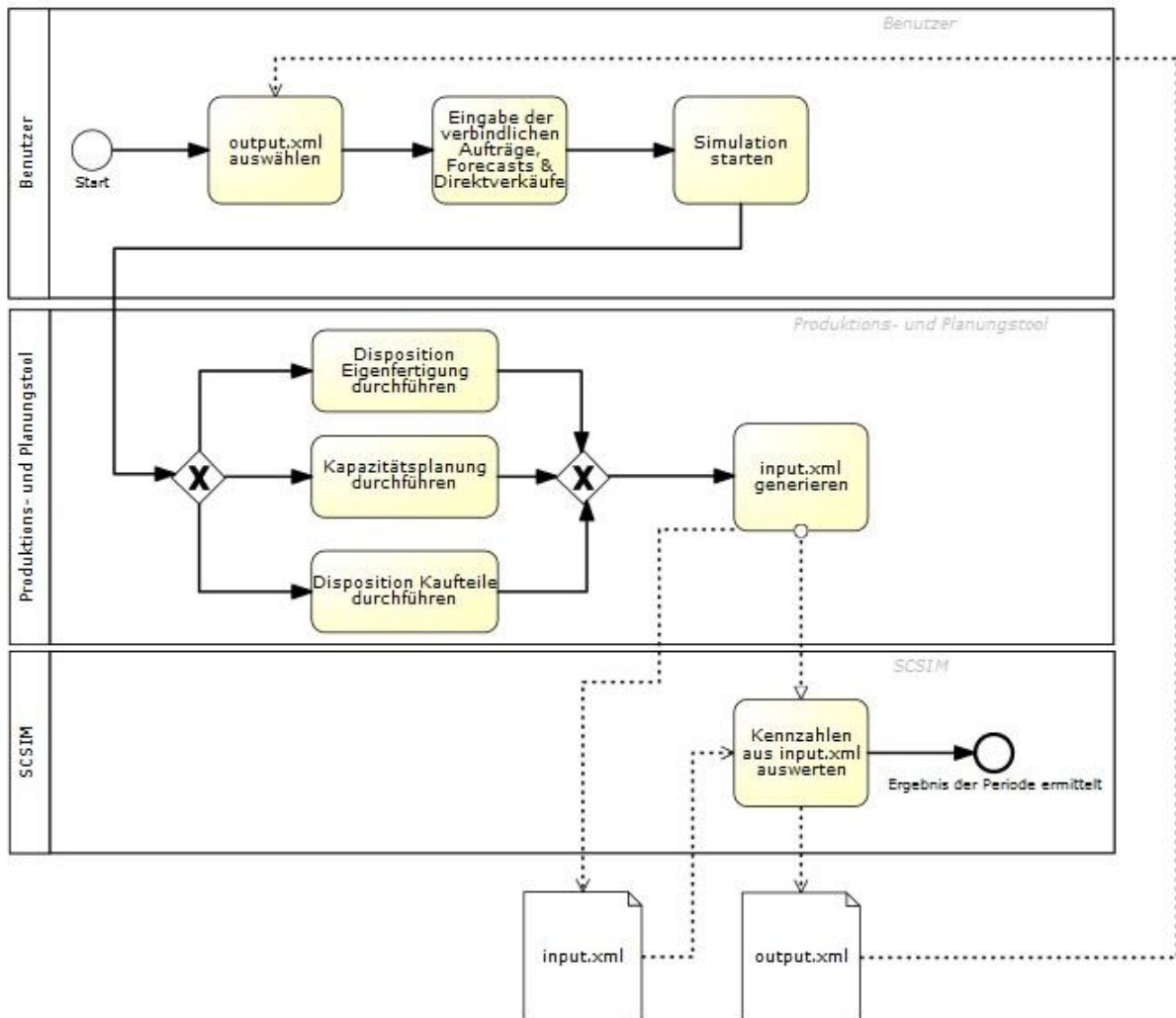
Die Grafik veranschaulicht, dass die Logistik-Aufgaben (grüner Kasten) nun nicht mehr vom Benutzer selbst bewältigt werden müssen, sondern diese vom PPT abgenommen werden. Der Benutzer gibt lediglich die Verbindlichen Aufträge sowie die Forecasts in das PPT ein und wählt die output.xml der vergangenen Periode aus. Das PPT errechnet nun die Kennzahlen und generiert eine input.xml die wiederum den Input für den SCSIM darstellt.

Eine genaue Anleitung zur Bedienung des PPT finden Sie unter 4.Bedienungsanleitung.

4. Bedienungsanleitung

Bei dem PPT handelt es sich um eine Webanwendung. Diese kann in jedem beliebigen Webbrowser ausgeführt werden.

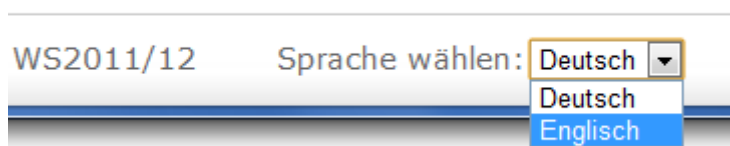
Den Ablauf und das Zusammenwirken von Benutzer, PPT und SCSIM soll das folgende BPMN deutlich machen:



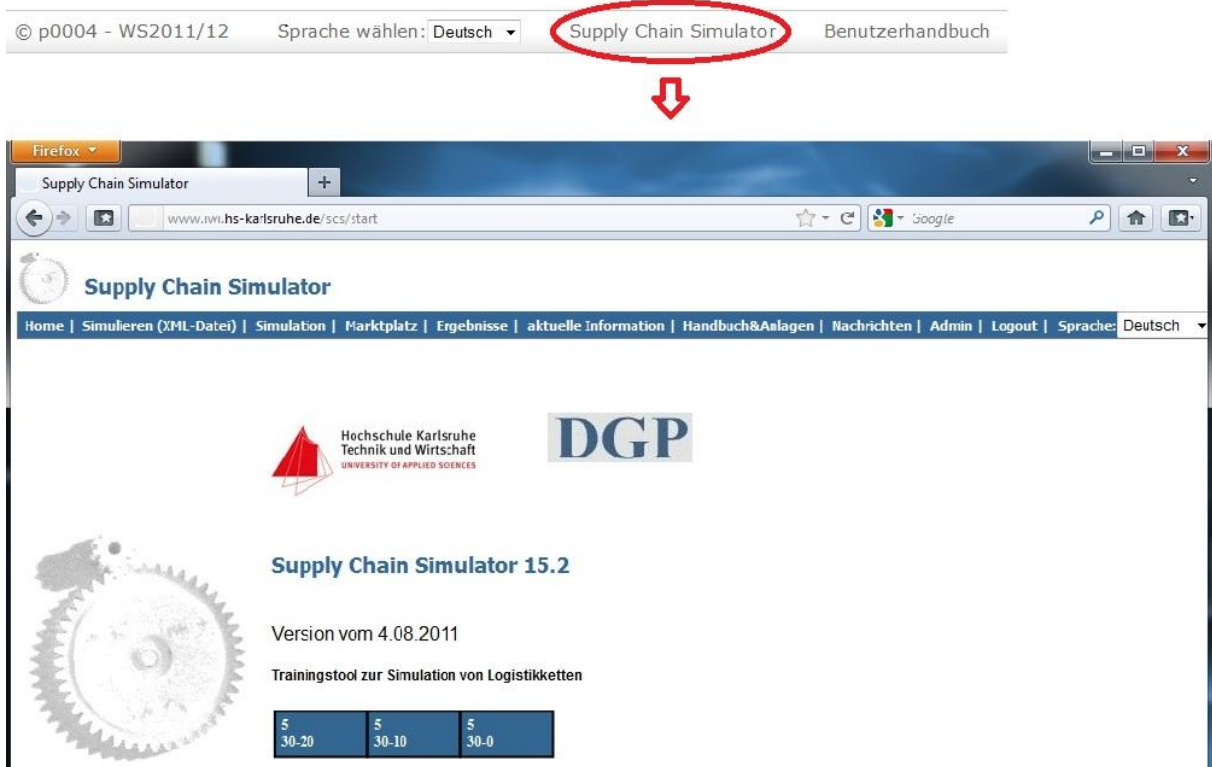
Nun kommen wir zur eigentlichen Bedienung des PPT. Die Startseite sieht folgendermaßen aus:



In der unteren Leiste finden Sie einige Optionen. Beispielsweise kann die Sprache auf Englisch umgestellt werden:



Durch Klick auf „Supply Chain Simulator“ gelangen Sie auf dessen die Website (www.scsim.de bzw. <http://www.iwi.hs-karlsruhe.de/scs/start>). Hier können Sie die mit dem PPT-Tool Simulierten Ergebnisse importieren.



Weiterhin steht Ihnen das Benutzerhandbuch zur Verfügung.



Kommen wir aber nun zur eigentlichen Anwendung. Durch Klicken auf den grünen Pfeil starten Sie das PPT.



Nachdem Sie die Simulation gestartet haben, werden Sie dazu aufgefordert eine xml-Datei hochzuladen. Bei der xml-Datei handelt es sich um eine output.xml, welche vom Supply Chain Simulator generiert wurde und das Ergebnis der vergangenen Periode enthält.

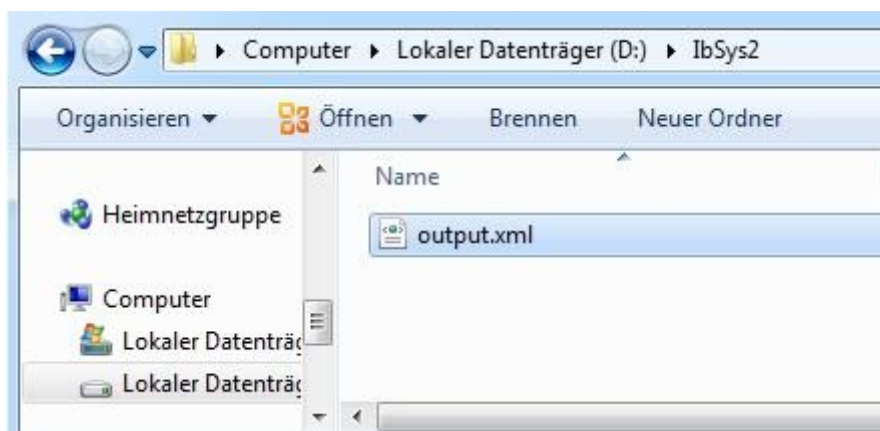
Klicken Sie auf *Durchsuchen* um die gewünschte xml-Datei auszuwählen.

Schritt 1: XML-Datei hochladen

init_result.xml



Auswahl der output.xml



Durch Klick auf den Vorwärts-Pfeil  gelangen Sie auf die nächste Seite.

Bei diesem Schritt werden die Marktinformationen in die dafür vorgesehenen Felder eingetragen. Geben Sie die verbindlichen Aufträge der kommenden Periode sowie die Forecasts der darauffolgenden Perioden für die Produkte P1, P2 und P3 ein. Falls vorgesehen, geben Sie zusätzlich jeweils Angaben zu Menge, Preis und Vertragsstrafe für die Direktverkäufe an.

IbSys II - Tool

Schritt 2: Marktinformationen eingeben

Periode	Verbindliche Aufträge	Prognosen				Direktverkauf		
	x	x+1	x+2	x+3	Menge	Preis	Vertragsstrafe	
P1	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0.0"/>	<input type="text" value="0"/>	
P2	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0.0"/>	<input type="text" value="0"/>	
P3	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0.0"/>	<input type="text" value="0"/>	

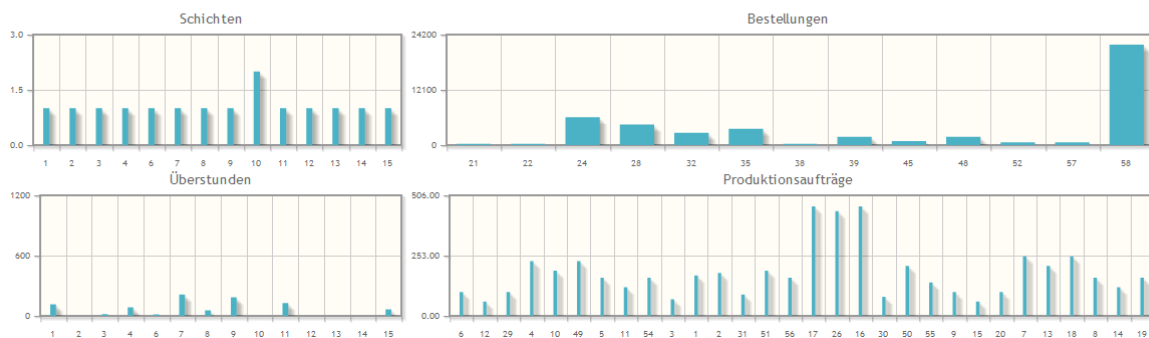
Konfiguration	
Prognoserechnung	Losgrößensplitting
Periode 1:	<input type="text" value="3.0"/> Los: <input type="text" value="100"/>
Periode 2:	<input type="text" value="2.0"/>
Periode 3:	<input type="text" value="1.0"/>
Produktionsfaktor:	<input type="text" value="0.85"/>



Grafische Darstellung der kalkulierten Werte:

IbSys II - Tool

Schritt 3: Berechnung überprüfen



5. Umsetzung

5.1 Eingesetzte Technologien

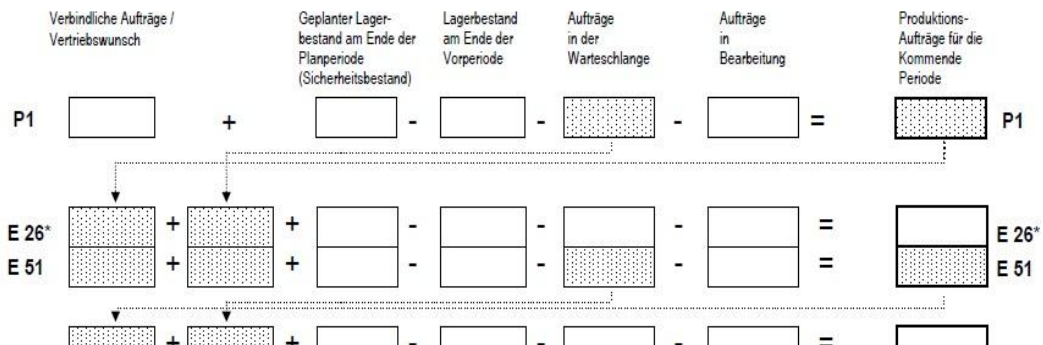
- Maven-Projekt
- Programmiersprache: J2EE
- XML-Handling: Java Architecture for XML-Binding (JAXB)
- Entwicklungsumgebung: Netbeans

5.2 Implementierung

5.2.1 Implementierung Disposition Eigenfertigung

Supply Chain Simulation

SCS



Die Berechnung der Produktionsaufträge orientiert sich stark an den Eingabefeldern:

```
double productionOrders = salesOrders
+ prevWaitingOrders[0] // Übertrag Produktionsaufträge
+ plannedStock          // Gewünschter Lagerbestand (Periodenende)
- getAmount()           // Lagerbestand
- waitingOrders[0]      // Aufträge in Warteschlange
- waitingOrders[1];     // Aufträge in Bearbeitung
```

5.2.2 Implementierung Kaufteildisposition

Produktionsprogramm									
Periode	1	2	3	4					
P1									
P2									
P3									

Nr. Kaufteil	Lieferfrist	Abweichung	Verwendung			Diskontmenge	Anfangsbestand in Per n	Bruttobedarf gemäß Produktionsprogramm				Bestellung		Bestand nach geplantem Wareneingang				
			P1	P2	P3			1	2	3	4	Menge	N=normal E=teil	2	3	4	5	
21	1,8	0,4	1 x			300	300											
22	1,7	0,4		1 x		300	300											
23	1,2	0,2			1 x	300	300											
24	3,2	0,3	7 x	7 x	7 x	6100	6100											
25	0,9	0,2	4 x	4 x	4 x	3600	3600											
27	0,0	0,0	2 x	2 x	2 x	1800	1800											

In den oberen Teil „Produktionsprogramm“ werden die Forecasts der kommenden Perioden eingetragen. Der *Bruttobedarf gemäß Produktionsprogramm* errechnet sich durch eine Matrixmultiplikation der Forecasts und der Anzahl der Verwendung in den Produkten P1, P2 und P3. Anschließend kann überprüft werden wie lange die im Lager verfügbaren Teile ausreichen um den Bedarf am Markt zu decken. Unter Berücksichtigung der Lieferfrist und der Abweichung wird entschieden ob ein Kaufteil bestellt werden muss, bzw. ob eine Normal- oder Eilbestellung vorliegt.

Die Methode `setOrderList()`

```

136 private void setOrderlist() {
137     Orderlist orderlist = new Orderlist();
138
139     List<double[]> orderInfoList = new ArrayList<double[]>();
140     orderInfoList.add(new double[]{21,1.8,0.4,1,0,0,300});
141     orderInfoList.add(new double[]{22,1.7,0.4,0,1,0,300});
142     orderInfoList.add(new double[]{23,1.2,0.2,0,0,1,300});

```

In der Liste **orderInfoList** werden Informationen in folgender Reihenfolge gesetzt:

{KaufteilNr, Lieferfrist, Abweichung, TV P1, TV P2, TV P3, Discountmenge}

(TV = Teileverwendung)

```

int period0 = (int)orderInfo[3] * resultDTO.getSalesOrdersP1() + (int)orderInfo[4]
int period1 = (int)orderInfo[3] * resultDTO.getForecastPlf1() + (int)orderInfo[4]
int period2 = (int)orderInfo[3] * resultDTO.getForecastPlf2() + (int)orderInfo[4]
int period3 = (int)orderInfo[3] * resultDTO.getForecastPlf3() + (int)orderInfo[4]

int warehousestock = resultDTO.getResult().getWarehousestock().getArticle().get
int futureinwardstockmovement = 0;
if (resultDTO.getResult().getFutureinwardstockmovement().getOrder() == null) {
    futureinwardstockmovement = resultDTO.getResult().getFutureinwardstockmovem
}
Order order = new Order();
if (period0 >= (warehousestock + futureinwardstockmovement)) {
    order.setArticle(BigInteger.valueOf(((int)orderInfo[0])));
    if ((warehousestock + futureinwardstockmovement + orderInfo[6]) <= period0)
        order.setQuantity(BigInteger.valueOf(((int)orderInfo[6] + period0)));
    } else {
        order.setQuantity(BigInteger.valueOf(((int)orderInfo[6])));
    }
    order.setModus(BigInteger.valueOf(4));
    orderlist.getOrder().add(order);
} else if (((period0 + period1) >= (warehousestock + futureinwardstockmovement))

```

int period0 – int period3:

Die Informationen TV1 – TV3 aus der Liste orderInfoList werden gemeinsam mit den zuvor im PPT als Marktinformationen eingegebenen Verbindlichen Aufträgen sowie Forecasts genutzt um den Bruttobedarf gemäß Produktionsprogramm zu berechnen (Matrixmultiplikation).

int warehousestock & int futureinwardstock:

In diese Variablen werden der aktuelle Lagerbestand und die zukünftigen Lagereingänge gespeichert.

If-Anweisung:

In der If-Anweisung wird nun geprüft wie lange der Lagerbestand gemeinsam mit den zukünftigen Lagereingängen den Marktbedarf decken kann. Kann dieser unter Berücksichtigung der Lieferzeit und Abweichung nicht entsprechend gedeckt werden, wird eine Bestellung durchgeführt und je nach zeitlicher Dringlichkeit der Modus auf Normal- oder Eilbestellung gesetzt.

5.2.3 Implementierung Kapazitätsplan

Periode ...	Sach-Nr.	Auftragsmenge	Arbeitsplatz														
Bezeichnung			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hinterrad	K E4											4	3				
	D E5											4	3				
	H E6											4	3				
Vorderrad	K E7											4	3				
	D E8											4	3				
...																	
...																	
komplett (cpl)	D P2					1											
	H P3					7											
Kapazitätsbedarf (neu)																	
Rüstzeit (neu)																	
Kap.bed. (Rückstand Vorperiode)																	
Rüstzeit (Rückstand Vorperiode)																	
Gesamt-Kapazitätsbedarf *)																	
Schichten und Überstunden :																	

Wenn die Produktionsaufträge für die kommende Periode feststehen, können diese in das Feld *Auftragsmenge* des jeweiligen Teils eingetragen werden. Es ist ersichtlich welches Erzeugnis an welchen Arbeitsplätzen gefertigt wird und auch, wie viel Minuten dort dafür aufgewendet werden. Nachdem für jedes Teil an den entsprechenden Arbeitsplätzen der Zeitbedarf ermittelt wurde kann der Kapazitätsbedarf jedes Arbeitsplatzes berechnet werden. Nun werden noch die Rüstzeiten miteinbezogen um den Gesamtkapazitätsbedarf zu ermitteln und schließlich kann anhand diesem

Wie dies im PPT umgesetzt wurde, zeigt die folgende Methode:

Die Methode setWorkingTimeList()

```
private void setWorkingtimelist() {
    Workingtimelist workingtimelist = new Workingtimelist();

    List<int[]> workstationInfo = new ArrayList<int[]>();
    // {workstation, setup time, article, process-time, article}
    workstationInfo.add(new int[]{1,20,49,6,54,6,29,6});
    workstationInfo.add(new int[]{2,30,50,5,55,5,30,5});
    workstationInfo.add(new int[]{3,20,51,5,56,6,31,6});
    workstationInfo.add(new int[]{4,30,1,6,2,7,3,7});
    workstationInfo.add(new int[]{6,15,16,2,18,3,19,3,20,3});
    workstationInfo.add(new int[]{7,20,10,2,11,2,12,2,13,2,14,2});
}
```

Es wird eine Liste *workingtimelist* angelegt in der folgende Informationen zu den Arbeitsplätzen gespeichert sind.

{ArbeitsplatzNr, Rüstzeit, Erzeugnis x, DLZ Erzeugnis x, Erzeugnis y, DLZ Erzeugnis y,...}

DLZ = Durchlaufzeit

```
// Setup time and setup events
int setupTime =workstation[1];
int setupEvents = resultDTO.getSetupEvents()[idx];

int total = machineTime + setupTime + setupEvents;

if (total <= 2400) {
    workingtime.setShift(BigInteger.ONE);
    workingtime.setOvertime(BigInteger.ZERO);
} else if (total > 2400 && total <= 3600) {
    workingtime.setShift(BigInteger.ONE);
    workingtime.setOvertime(BigInteger.valueOf((total - 2400) / 5));
} else if (total > 3600 && total <= 4800) {
    workingtime.setShift(BigInteger.valueOf(2));
    workingtime.setOvertime(BigInteger.ZERO);
} else if (total > 4800 && total <= 6000) {
    workingtime.setShift(BigInteger.valueOf(2));
    workingtime.setOvertime(BigInteger.valueOf((total - 4800) / 5));
} else {
    workingtime.setShift(BigInteger.valueOf(3));
    workingtime.setOvertime(BigInteger.ZERO);
}
```

int setupTime:

In dieser Variablen wird die Rüstzeit gespeichert.

int setupEvents:

Wie häufig an einem Arbeitsplatz gerüstet wird ist nur schwer vorhersagbar. Daher wurden festgelegte Werte genommen die an Vergangenheitswerten orientiert sind. Änderungen können jederzeit in einer Properties-Datei vollzogen werden um eventuelle Anpassungen vorzunehmen.

int total:

Die Variable total, speichert den Gesamtkapazitätsbedarf des entsprechenden Arbeitsplatzes.

If-Anweisung:

Im ersten Teil der If-Anweisung wird geprüft, ob der Gesamtkapazitätsbedarf bei einem Wert kleiner oder gleich 2400 Minuten liegt. Ist dies der Fall, werden eine Schicht und keine Überstunden aufgetragen. Dies wiederholt sich im dritten und im fünften Teil der Anweisung mit dem Unterschied, dass die Schicht auf 2 bzw. auf 3 gesetzt wird.

Das Prinzip des zweiten sowie des vierten Teils der Anweisung liegt darin, dass geprüft wird ob der Gesamtkapazitätsbedarf zwischen zwei Werten liegt bei dem Überstunden eingeplant werden müssen.