

# Entwicklung einer performanten zentralen Produktionsplanung zu Vergleichs- und Integrationszwecken

Pascal Schumann

HTW Dresden

20. Dezember 2019

- ① Begrüßung
- ② Einleitung
- ③ Anforderungen
- ④ Umsetzung
- ⑤ Performancestudie
- ⑥ Zusammenfassung

- In MA Krockert und Seifert entwickelte „Sich selbst organisierende Produktion“ (SSOP) war eine zentrale Produktionsplanung (ZPP) enthalten.
- ZPP der SSOP nicht leistungsfähig genug (Laufzeit)
- Fehlen einer einheitlichen Simulationsplattform, um ggf. hybride Lösungen bauen zu können

→ Wunsch nach neuer Umsetzung mit expliziten Anforderungen an die Performance

- Entwicklung einer performanten zentralen Produktionsplanung (ZPP)
- Vergleichbarkeit der ZPP mit der SSOP → Integrationsmöglichkeit der ZPP in die SSOP
- Hinreichend kurze Laufzeit aufweisen
- Funktional korrekt
- Gute Struktur für Erweiterbarkeit bzw. Ersetzbarkeit von Komponenten

# Anforderungen

Die ZPP muss folgende Anforderungen erfüllen:

- MRP-Lauf (MRP II)
- Neuplanung unter Verrechnung von Rückmeldungen
- Verwendung der Simulation der SSOP
- Gute Struktur → Anwendung der Objektgymnastik
- Empirischen Studie zur Überprüfung der Performanz (Performancestudie)
- Integration der ZPP in die SSOP

Der MRP-Lauf der ZPP muss folgende Anforderungen erfüllen:

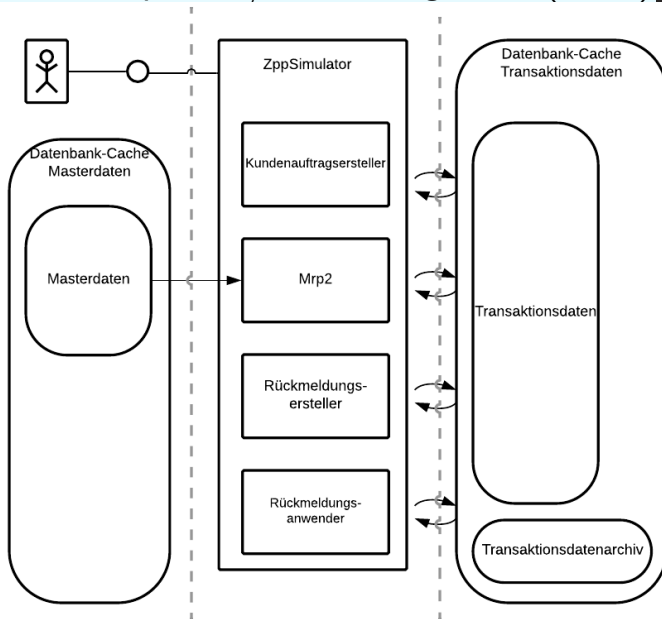
- Stücklistenauflösung mit Produktionsauftrags- und Bestellungsanlage
- Schnittstelle für Losgrößenbildung online und offline
- Durchlaufterminierungen r-v-r, online und offline
- Automatischer Kapazitätsabgleich aufgrund des „Abstreifermodells“
- Maschinenbelegung: Giffler-Thompson-Algorithmus und Schlupfzeitregel als Prioritätsregel

# Umsetzung



1. Generierung der Kundenaufträge
2. MRP II
  - 2.1 Materialbedarfsplanung
  - 2.2 Kapazitätsplanung gegen unbegrenzte Kapazität:  
Durchlaufterminierungen (rückwärts-vorwärts-rückwärts)
  - 2.3 Kapazitätsplanung gegen begrenzte Kapazität: Maschinenbelegung
3. Simulation der Produktion und Rückmeldungen der Operationen  
(Start- und Endzeiten)
4. Anwendung der Rückmeldungen

# Architektur Component/Block Diagramm (TAM)



- NP-schweres Optimierungsproblem → Verwendung einer Heuristik
- Heuristik findet eine gute Lösung → keine Aussage über Entfernung vom Optimum möglich
- Verwendete Heuristik: Giffler-Thompson-Algorithmus
- GT-Algorithmus benötigt Prioritätsregel (Schlupfzeitregel)

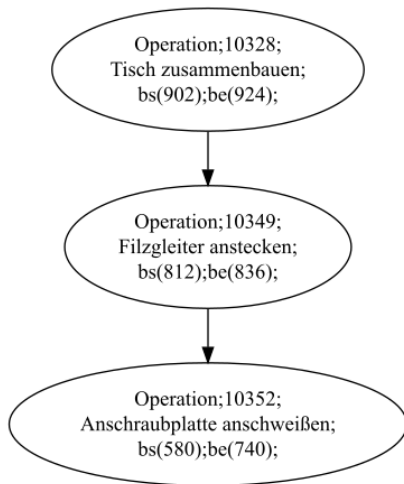


Abbildung: Der Operationsgraph des Tischbeispiels als gerichteter Graph

```
1 // Korregiere LS jeder Maschine für alle Operationen der letzten Zyklen
2 // Bestimme initiale Menge: S := alle Blätter von og
3 // Startzeit = Startzeit aus RT für alle o aus S
4 while S not empty do
5     Berechne  $d(o) = t(o) + p(o)$  für alle o aus S
6      $d\_min := \min\{ d(o) \mid o \text{ aus } S \}$ 
7      $o\_min := o(d\_min)$ 
8     Bilde Konfliktmenge  $K = \{ o \mid o \text{ aus } S \text{ UND}$ 
9         Maschinengruppe(o) == Maschinengruppe(o_min) UND  $t(o) < d\_min \}$ 
10    while K not empty do
11        foreach machine in Maschinengr.(o_min) geordnet nach LS ASC
12            // Entnehme Operation mit höchster Priorität (o1) aus K
13            und plane o1 auf maschine ein
14            // Korrigiere  $t(o1)$ , falls  $ls > t(o1)$  || benötigte Material
15            später bereitsteht
16        // Setze Startzeit aller übrigen o aus K auf die Endzeit der
17        zuletzt eingeplanten Operationen
18        foreach eingeplanter Operationen
19             $t(o) := d(o1)$  für alle o aus N(o1)
20            Entferne o1 aus og
21    S := alle Blätter von og
```

## 9 Strukturregeln von Jeff Bay:

- 1 | Verwende höchstens eine Einrückungsebene pro Methode.
- 2 | Nutze nicht das „Else“-Schlüsselwort.
- 3 | Alle Primitive und Zeichenketten kapseln.
- 4 | Nutze nur einen Punkt pro Anweisung (statement).
- 5 | Bezeichner nicht abkürzen.
- 6 | Alle Entitäten kleinhalten.
- 7 | Eine Klasse soll höchstens zwei Instanzvariablen verwalten.
- 8 | Container (Collection) als Instanzvariable in eine eigene Klasse auslagern.
- 9 | Nutze keine „Getters“, „Setters“ und Instanzvariable außerhalb der Klasse.

Umsetzung: Demo für Regel 3 und 8

## Integrations- und Unittests für

- MRP-Lauf
- Rückmeldungen der Operationen
- Anwendung der Rückmeldungen
- Objektgymnastik

✓ release/0.3 adapt performance study parameters

↪ #680 passed

↪ Commit 120d38d ↪

🕒 Ran for 35 min 27 sec

↪ Compare 0.3 ↪

📅 26 days ago

↪ Branch release/0.3 ↪

👤 Pascal

🔧 </> Mono: none C#

🖨️ AMD64

```
.....Test Run Successful.
388 Total tests: 78
389     Passed: 73
390     Skipped: 5
391 Total time: 30.5087 Minutes
392 The command "dotnet test Zpp/Zpp.csproj
393 0.
394
395 Done. Your build exited with 0.
```



# Performancestudie

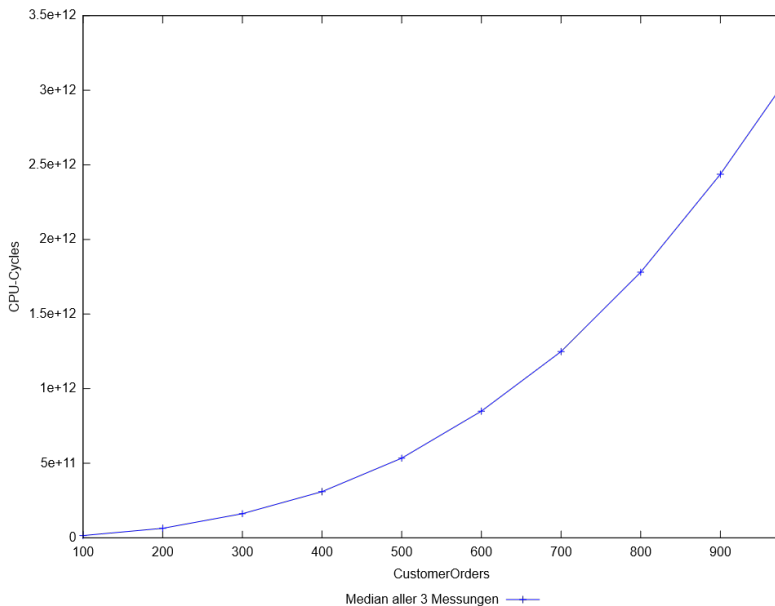
Genauer: Empirische Studie zur Überprüfung der Performance der ZPP  
(Performancestudie)

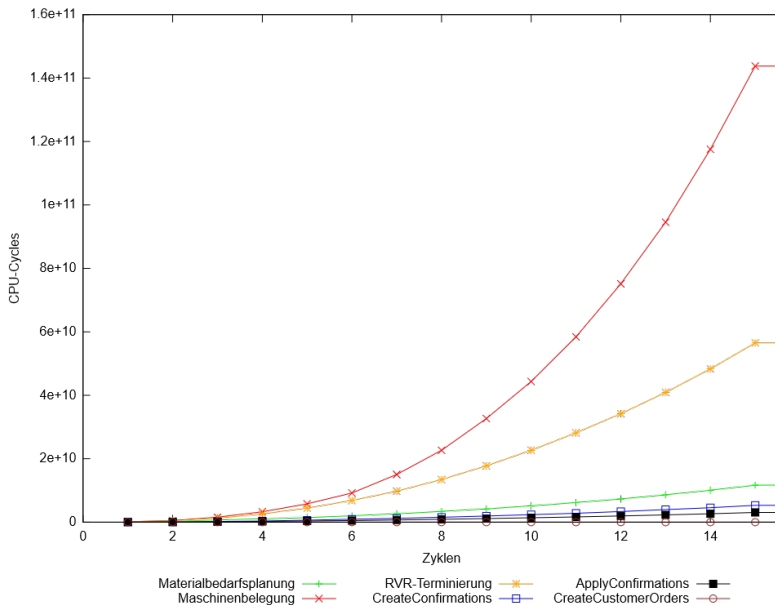
Betrachtung der Fälle:

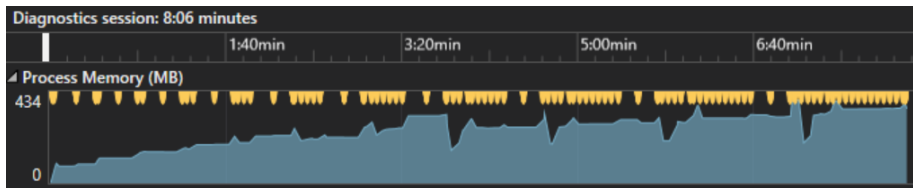
1. Laufzeit/Speicherbedarf zwischen 100 und 1000 Kundenaufträgen wg. Laufzeitentwicklung
2. Laufzeit/Speicherbedarf pro Simulationszyklus bei 500 Kundenaufträgen wg. Komponentenverteilung

Bedingungen:

- Maximale Simulationsdauer 20160min
- Ein Simulationszyklus: 1440min → mind. 14 Zyklen
- Stücklisten und Operationen der Trucks (2) aus SSOP wg. Vergleichbarkeit mit SSOP
- Trucks: 3-stufige Stückliste mit mind. 25 Operationen







# Zusammenfassung

Die ZPP muss folgende Anforderungen erfüllen:

MRP-Lauf (MRP II)

Neuplanung unter Verrechnung von Rückmeldungen

Verwendung der Simulation der SSOP

Gute Struktur durch Objektgymnastik

Empirischen Studie zur Überprüfung der Performanz (Performan-  
cestudie)

Integration der ZPP in die SSOP

NF



Der MRP-Lauf der ZPP muss folgende Anforderungen erfüllen:

- Stücklistenauflösung mit Produktionsauftrags- und Bestellsanla-  
ge ✓
- Schnittstelle für Losgrößenbildung online und offline ✓
- Durchlaufterminierungen r-v-r, online und offline ✓
- Automatischer Kapazitätsabgleich aufgrund des „Abstreifermodells“ ✗
- Maschinenbelegung: Giffler-Thompson-Algorithmus und Schlupfzeit-  
regel als Prioritätsregel ✓



Erreicht:

- Leistungsfähige ZPP
- Grundlage für hybride Planungslösungen gelegt

Ausstehende Tätigkeiten:

- Automatischer Kapazitätsabgleich aufgrund des „Abstreifermodells“
- Integration der ZPP in die SSOP ✓
- Validierung der ZPP (Feldversuche)

Vielen Dank.