

Gruppe kass3

# Handbuch PPS

IBSYS II – SS2020

Leon Fuchs, Tim Häusler, Constantin Krämer, Luis Wettach

## Inhalt

1.	Einleitung .....	2
2.	Ausgangssituation .....	2
3.	Problembeschreibung .....	3
4.	Lösungsansatz .....	3
4.1	Aufbau des Backends .....	5
4.2	Aufbau des Frontends .....	8

## 1. Einleitung

Die Produktionsplanung – eine der wichtigsten Bestandteile Ihres Unternehmens. Häufig ist sie zeitaufwändig und fehleranfällig. Werden die Planungsschritte nicht ordnungsgemäß und sorgfältig durchgeführt, kann dies zu einem erheblichen Wettbewerbsnachteil führen. Mittlerweile setzen fast alle erfolgreichen Unternehmen Produktionsplanungstools, wie das unsere ein. Dieses automatisiert die Produktionsprogrammplanung, Materialbedarfsplanung und die Produktionsprozessplanung. Mit der benutzerfreundlichen Bedienung erreichen Sie im Handumdrehen exzellente Ergebnisse und verschaffen sich einen erheblichen Wettbewerbsvorteil.

## 2. Ausgangssituation

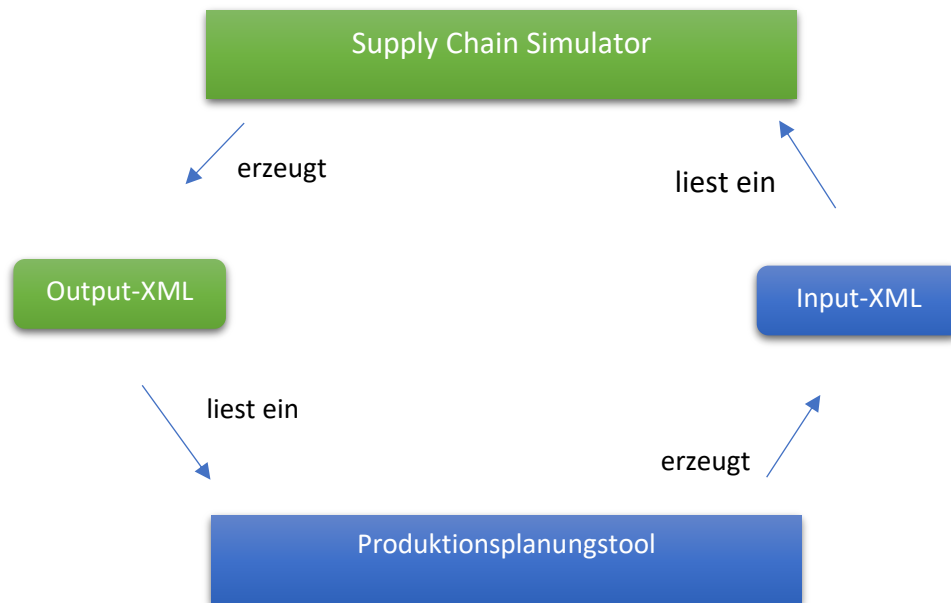
Im Rahmen der Veranstaltung IBSYS II soll ein Produktionsplanungstool entwickelt werden. Dieses soll alle relevanten Daten für das Planspiel „Supply Chain Simulator“ automatisiert ermitteln. In dem Planspiel werden drei verschiedene Varianten von Fahrrädern (Herren-, Damen-, und Kinderfahrräder) gefertigt und vertrieben. Diese werden in der simulierten Produktion an 14 Arbeitsplätzen, aus Eigenfertigungs- und Kaufteilen hergestellt. Die Produktion der Eigenfertigungsteile und Teile muss in der Eigenfertigungsdisposition geplant werden. Die in der Eigenfertigung benötigten Fertigungs- und Rüstzeiten müssen in der Kapazitätsplanung der Arbeitsplätze berücksichtigt werden. Anhand des Produktionsprogramms muss der Einkauf von Kaufteilen geplant werden.

Die Planung erfolgt periodisch, eine Periode entspricht einer Arbeitswoche á 5 Tagen. Das Simulationstool nimmt folgende Parameter in Form eines XML-Flies (Input-XML) entgegen:

- Vertriebswunsch für die kommende Periode
- Eigenfertigungsdisposition
- Kapazitätsplan für Arbeitsplätze
- Kaufdisposition

Das Ergebnis der Simulation wird ebenfalls in Form eines XML-Files zu Verfügung gestellt, dieses beinhaltet folgende Nenngrößen:

- aktueller Lagerbestand
- Lagerzugang in der simulierten Periode
- Lagerzugang in kommenden Perioden
- Leerzeitkosten
- Warteschlangen Arbeitsplätze
- Warteschlange Material
- Aufträge in Bearbeitung
- Abgeschlossene Aufträge
- Ergebnis der simulierten Periode



### 3. Problembeschreibung

Wie bereits beschrieben erfolgt die Planung auf Grundlage des Output-XML und der Prognose des Vertriebs. Bei der Planung der Produktion sind folgende Planungsschritte und Restriktionen zu berücksichtigen:

- Erstellung eines Produktionsprogramms: anhand der Prognosedaten und ggf. Direktverkäufen muss ein Produktionsprogramm für die kommenden Perioden festgelegt werden. Das Produktionsprogramm soll für eine konstante Auslastung der Produktion sorgen, damit bei geringer Nachfrage keine Unterkapazität und bei hoher Nachfrage keine Überkapazitäten entstehen.
- Auflösung der Warteschlangen: Falls in der Vorperiode in der Kapazitätsplanung nicht genügend Zeit eingeplant wurde oder Teile für die Fertigung fehlen, kann es vorkommen, dass Produktionsaufträge nicht vollständig abgeschlossen werden konnten. In diesem Fall müssen die unfertigen Produktionsaufträge in der kommenden Periode berücksichtigt werden.
- Planung der Eigenfertigung: anhand des Produktionsplans für die kommende Periode und des Sicherheitsbestands für Eigenfertigungsteile wird die Disposition der Eigenfertigungsteile erstellt. Hierbei müssen auch die nicht abgeschlossenen Produktionsaufträge der vorausgegangenen Periode berücksichtigt werden.
- Kapazitätsplanung: Aus der Disposition der Eigenfertigung leitet sich die Planung der Fertigungskapazitäten ab. Hierfür wird für jeden Arbeitsplatz die Fertigungszeit und die Rüstzeit für jedes Produktionsteil errechnet. Auch hier müssen wieder die Produktionsrückstände aus der vorhergegangenen Periode berücksichtigt werden. Der Kapazitätsplan gibt an, wie viele Sichten und wie viele Überstunden pro Arbgeleistet werden müssen.
- Planung der Kaufdisposition: Abhängig von dem Produktionsplan und den Produktionsrückständen der Vorperiode muss die Kaufdisposition geplant werden. Anhand dieser Parameter und der Lieferfristen und Mengen müssen die optimalen Bestellzeitpunkte und die optimalen Bestellmengen ermittelt werden. Bei optimalen Bestellmengen und Bestellzeitpunkten sind Lieferkosten und Lagerhaltungskosten minimal. Die Lieferung erfolgt zu

- Restrektionen zu den Kapazitäten der Arbeitsplätze: Insgesamt sind 14 Arbeitsplätze vorhanden. An jedem Arbeitsplatz können bis zu drei Schichten á 2400 min am Tag geplant werden. Je nach Schicht fallen Kosten pro Arbeitsminute an. Die erste Schicht ist die günstigste, die letzte Schicht ist die teuerste. Es können auch Überstunden eingeplant werden, wenn die zu erwartende Kapazität an einem Arbeitsplatz bis zu 50% über max. einer Schicht liegt. Nach der dritten Schicht können keine zusätzlichen Überstunden geplant werden.
- Restrektionen zu Lagerhaltungskosten: Lagerkosten errechnen sich aus dem durchschnittlichen Lagerwert einer Periode, sie entsprechen 0,6% des Lagerwerts. Steigt der Lagerwert über den Betrag von 250.000€ so muss eine zweite Lagerhalle angemietet werden, diese verursacht fixe Kosten in Höhe von 5000€. Es empfiehlt sich daher den Lagerwert stets unter 250.000€ zu halten.
- Bestelltypen und Bestellkosten: Es sollen 2 verschiedene Bestelltypen unterstützt werden. Normalbestellungen und Eilbestellungen. Normalbestellungen verursachen normale Lieferkosten, je nach Produkt betragen diese 50€, 75€ oder 100€. Bei Normalbestellungen erfolgt die Lieferung zu den angegebenen Lieferzeiten mit Normalverteilten Abweichungen. Bei Eilbestellungen erfolgt die Lieferung in der halben Lieferzeit, dabei gibt es keine Lieferzeitabweichungen. Bei Eilbestellungen betragen die Lieferkosten je nach Produkt 500€, 750€ oder 1000€. Neben der Eilbestellung gibt es auch noch die Just in Time Bestellung. Diese ermöglicht die Bestellung in 0,3 der normalen Lieferzeit ohne Abweichung es entstehen aber deutlich höhere Kosten.

## 4. Produktumfang

Mit dem Erwerb des Tools erhalten Sie direkten Zugriff auf die Webapp und das Benutzerhandbuch. Eine Inbetriebnahme des Tools Ihrerseits ist nicht notwendig, das übernehmen wir für Sie. Trotzdem haben Sie die Möglichkeit den Quellcode einzusehen.

Bei Fragen oder Problem bezüglich des Produkts zögern Sie nicht unser Supportteam zu kontaktieren, wir stehen Ihnen Werktags von 9 bis 17 Uhr für Ihrer Fragen zur Verfügung.

Die Anwendung ist unter folgendem Link zu erreichen:

Der Quellcode ist unter folgendem Link einzusehen:

## 5. Bedienungsanleitung

Das Tool ist wie bereits erwähnt unter folgendem Link zu erreichen: [<Link>](#).

Im folgenden Abschnitt werden Sie Schritt für Schritt durch alle Planungsschritte geleitet, hierbei werden Ihnen alle Funktionen des Tools und alle Eingabemasken erläutert. Im Allgemeinen führt Sie das Tool, gemäß der betrieblichen Reihenfolge, nach und nach durch alle Planungsschritte. Sie haben stets die Möglichkeit kalkulierte Werte händisch zu ändern, um so die Planung zu optimieren. Ebenso können Sie Ihre vorhergehenden Schritte korrigieren und zu vorangegangenen Planungsabschnitten zurückspringen.

### 5.1 Sprachauswahl

Das Tool steht Ihnen in den Sprachen Deutsch und Englisch zu Verfügung. Die Sprache lässt sich immer über das Drop-down Menü oben rechts, neben dem Homebutton ändern.

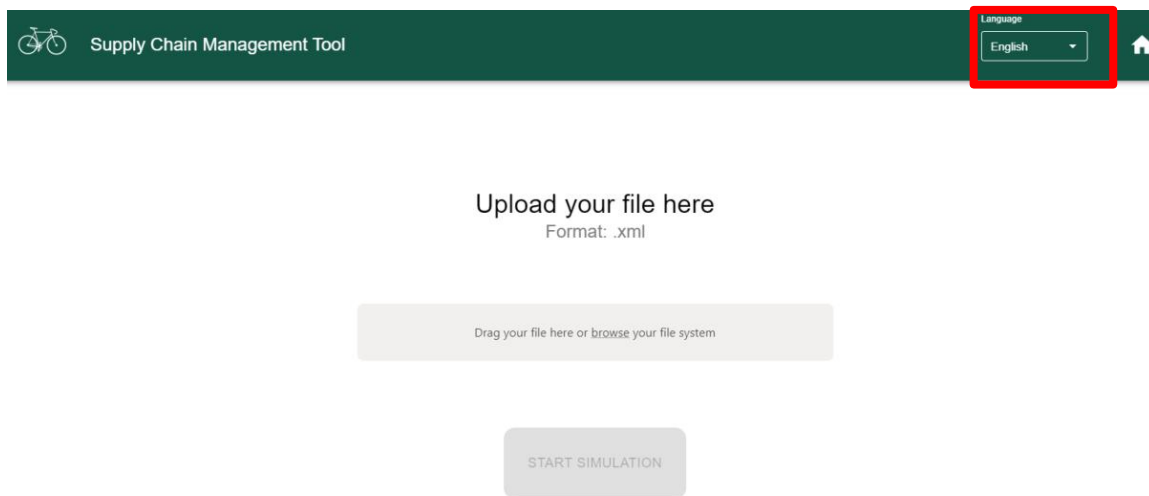


Abbildung 1

## 5.2 Schritt für Schritt Anleitung

### 1. Einlesen der Ergebnisse der Vorperiode

Im ersten Schritt müssen die Ergebnisse der Vorperiode eingegeben werden. Die dazugehörigen Ergebnisse liefert das Simulationstool in Form einer XML-Datei. Diese XML-Datei können sie entweder per Drag and Drop, in das in Abbildung 2 markierte Feld ziehen. Alternativ gelangen Sie durchklicken in das Feld in der Datei-Explorer, über diesen können Sie dann die XML-Datei in Ihrem Dateisystem suchen und auswählen. Mit dem Klicken auf „OK“ wird die Datei hochgeladen.

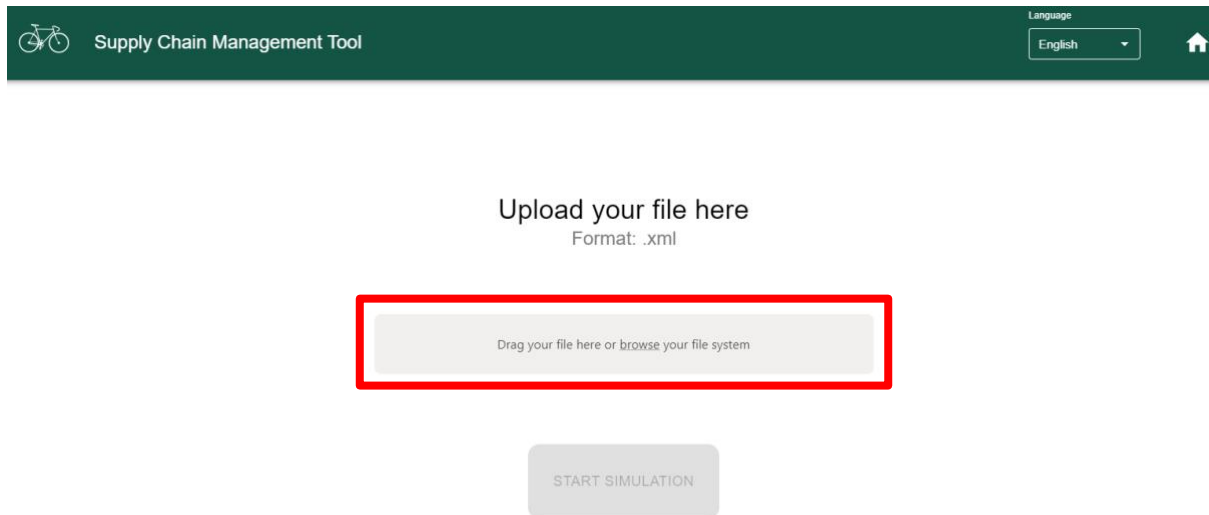


Abbildung 2

## 6. Technische Dokumentation

Die Software basiert auf einer klassischen Server-Client-Architektur, die untereinander mithilfe des leichtgewichtigen, asynchronen HTTP-Protokolls kommunizieren. Es folgen Informationen zum strukturellen Aufbau des Back- und Frontends.

### 6.1 Aufbau des Backends (Server)

#### 6.1.1 Technologien

Das Backend basiert auf dem Webframework „ASP.NET Core“ (Version 3.1) von Microsoft, welches heutzutage, aufgrund von großer Funktionalitätsvielfalt und hoher Performanz, trotz leichtgewichtigen Binaries, sehr weit in der Webentwicklung verbreitet ist. Mithilfe des in C# implementierten Programmcodes, welcher einsteigerfreundlich ist und leichte Verständlichkeit, auch für nicht Entwickler bietet, wird eine RESTful Schnittstelle realisiert, die alle benötigten HTTP Endpunkte für die Simulation bereitstellt. REST ist ein, in der Webentwicklung weit verbreitetes, Programmierparadigma für verteilte Systeme, insbesondere für Webservice, das üblicherweise die benötigten CRUD (Create, Read, Update, Delete) Operationen für einen Webclient bereitstellt.

## 6.1.2 Architektur

Bezüglich der Architektur folgt das Backend dem Standardaufbau einer typischen ASP.NET Applikation. Diese sieht vor, dass die benötigten Endpunkte einer Domäne in einem sogenannten Controller definiert werden. Diese Definitionen enthalten unter anderem die URI, sowie die erlaubte HTTP-Methode, über diese sie zu erreichen sind. Des Weiteren verweisen sie auf einen Service, der die die entsprechende Anwendungslogik verkapselt, um die Anfrage mit den empfangenen Daten zu verarbeiten und eine berechnete Antwort für den Client anfertigt. Diese wird dann an den Controller zurückgegeben und an den anfragenden Client gesendet.

Im konkreten Fall dieses Tools stehen folgende Endpunkte im „SimulationController“ bereit, um den Client mit allen benötigten Simulationsschritten zu bedienen:

Endpunkt	URI	HTTP-Methode	Beschreibung
UploadLastPeriodResults	simulation/results-input	POST	Empfängt die Ergebnisse der Vorperiode vom Client und persistiert sie im Arbeitsspeicher
Start	simulation/start	POST	Empfängt den Vertriebswunsch und Forecast, übergibt diese an den Eigenfertigungs-Dispositions-Service, welcher die Eigenfertigungsdispositionen berechnet und sendet die diese an den Client zurück
UpdateDispoEfP1	simulation/update-dispo-ef/p1	PUT	Empfängt die nachbearbeiteten Sicherheitsbestände für die Disposition Eigenfertigung P1 und berechnet daraus die neuen Eigenfertigungsdispositionen
UpdateDispoEfP2	simulation/update-dispo-ef/p2	PUT	Empfängt die nachbearbeiteten Sicherheitsbestände für die Disposition Eigenfertigung P2 und berechnet daraus die neuen Eigenfertigungsdispositionen
UpdateDispoEfP3	simulation/update-dispo-ef/p3	PUT	Empfängt die nachbearbeiteten Sicherheitsbestände für die Disposition Eigenfertigung P3 und berechnet daraus die neuen Eigenfertigungsdispositionen
GetKapazitaetsPlaene	simulation/kapa-plan	GET	Leitet die Anfrage an den Kapazitätsservice weiter, welcher den Kapazitätsplan berechnet und an den Controller zurückgibt, woraufhin dieser ihn an den Client sendet



GetKaufDispos	simulation/kauf-dispo	GET	Leitet die Anfrage an den Kauf-Dispositions-Service weiter, welcher die Kaufteiledisposition berechnet und an den Controller zurückgibt, woraufhin dieser ihn an den Client sendet
UpdateKaufDispo	simulation/update-kauf-dispo	PUT	Empfängt die nachbearbeiteten Stückzahlen für die Bestellung der Kauteile und berechnet daraus die neuen Kaufteiledispositionen
ConvertToXml	simulation/convert-to-xml	POST	Empfängt die Input Datei für um die Simulation zu starten im JSON Format, wandelt diese IN XML und gibt sie an den Client zurück

### 6.1.3 Datenhaltung

Da keinerlei Daten über den Simulationsprozess hinaus persistiert werden müssen, wurde bewusst auf eine Datenbank verzichtet. Es ist völlig ausreichend die Daten im Arbeitsspeicher zu halten, denn die für die Simulation relevanten Daten werden in der ausgegeben XML-Datei gespeichert und sind für eine weitere Planung irrelevant. Hierzu wurden alle Stammdaten aus den Excel Dateien in CSV Dateien gewandelt und ein Parser implementiert, der die Datensätze aus den CSV-Dateien in entsprechende C# Klassen wandelt. Dies passiert beim Start des Servers, woraufhin die Daten bis zum Abschalten des Servers im Arbeitsspeicher zur Verfügung stehen. Dadurch wurde nicht nur die Erstellung und Wartung einer Datenbank gespart, sondern auch die eine deutlich höhere Performanz bezüglich der Zugriffszeiten erreicht, da Arbeitsspeicherzugriffe um einiges schneller sind als Datenbankzugriffe. Um das Parsen der Ergebnisse der Vorperioden zu ermöglichen wurden außerdem entsprechende XSD Dateien erzeugt, die das Schema der XML-Datei abbilden. Mithilfe dieser XSD Dateien konnten entsprechende C#-Klassen generiert werden, welche die zu verarbeitenden Daten im passenden Datenschema halten.

## 6.2 Aufbau des Frontends (Client)

- Eingabemöglichkeiten
- Eingabevalidierung
- Kommunikation mit dem Backend
- Folgedialog

Beschreibung der einzelnen Eingabemasken und der