# Wstęp

## Wprowadzenie do problemu

Efektywne zarządzenie rzeczywistym łańcuchem dostaw nieoderwalnie związane jest z szeregiem procesów planowania. Przed każdym przedsiębiorstwem produkcyjno-dystrybucyjnym staje zadanie takiego sterowania przypływem materiałów w sieci, by zapewnić możliwie wysoki poziom obsługi klienta przy jednoczesnym minimalizowaniu zapasów oraz kosztów związanych ze składowaniem i transportem materiałów. Są to jedne z wielu wskaźników efektywności (ang. KPI[[1]](#footnote-1)), które pozwalają ocenić to, czy łańcuch dostaw zarządzany jest w sposób zbliżony do optymalnego.

Ilość danych, które należy przetworzyć i przeanalizować procesie planowania sprawia, że nieodzowne jest wykorzystywanie zaawansowanych systemów komputerowych. Praktycznie niemożliwa jest jednak pełna automatyzacja tego procesu. Liczba wyjątków występująca ze względu na różnorodność produktów, zróżnicowana struktura łańcuchów dostaw w sieci, a także dynamika branży FMCG[[2]](#footnote-2) uniemożliwiają potraktowania problemu dystrybucji jako klasycznego zadania optymalizacyjnego.

Konieczne staje się zatem rozpoznanie typowych problemów, które pojawić mogę się na ścieżce planowania produkcji i dystrybucji artykułów FMCG, a także wypracowania określonych strategii mitygowania ryzyka możliwych do wdrożenia w środowisku biznesowym. Z oczywistych względów nie jest wskazane, aby rozwijać i wypracowywać rozwiązania takich problemów wyłącznie w sposób praktyczny. Niesie to ze sobą znaczące ryzyko poniesienia strat pieniężnych, a także obarczone jest koniecznością długiego oczekiwania na rezultaty podjętych decyzji. Z tego powodu autor niniejszej pracy zdecydował się na wykorzystanie symulacji komputerowej jako narzędzia do badań. Wiąże się to również z szeregiem innych korzyści – możliwości modelowania i odwzorowania dowolnych łańcuchów dostaw, czy przeprowadzania wielokrotnych testów i badania wrażliwości systemu na zmiany parametrów jego poszczególnych ogniw.

Jakość wyników symulacji komputerowej jest jednak silnie zależna od zaimplementowanego w niej modelu i jego zbieżności z rzeczywistością. Aby pozostać możliwie blisko realiów biznesowych w pracy wykorzystane zostaną dane historyczne rzeczywistego przedsiębiorstwa X funkcjonującego w branży FMCG. Składać się na nie będą dane dotyczące popytu i sprzedaży, struktura łańcuchów dostaw, a także ograniczenia związane z wytwarzaniem poszczególnych produktów.

## Przegląd literatury

Problem modelowania łańcucha dostaw oraz badania przepływu materiałów w sieci nie jest nowy. Powstało wiele publikacji wpisujących się w tę tematykę, jednak żadna z nich nie wyczerpuje dostatecznie tematu wykorzystywania symulacji komputerowej w celu odnalezienia dobrych praktyk w zarządzaniu łańcuchem dostaw. Co więcej, w publikacjach można zaobserwować pewną polaryzację. Pierwsza grupa prac to te skupione wokół tworzenia jak najdokładniejszych modeli matematycznych łańcucha dostaw, które pozbawione są kontekstu biznesowego. Przeciwnym biegunem są zaś publikacje magazynów branżowych, kładące akcent przede wszystkim na aspekty biznesowe, które dowodzą słuszności stawianych w nich tez poprzez odwołania do rzeczywistych przedsiębiorstw i wdrożonych w nich rozwiązań oraz efektu ich implementacji.

*Modeling Suppy Chain Dynamics: A Multiagent Approach* (Swaminathan, Smith i Sadeh 1998) opisuje podstawy tworzenia modeli łańcucha dostaw w oparciu elementy dwóch klas – strukturalnej (odzwierciedlającej ogniwa łańcucha) i kontrolnej (reprezentującej przepływ popytu, materiałów i informacji). O ile publikacja stanowi dobry punkt wyjściowy dla tworzenia symulacji w oparciu o system wieloagentowy, nie dostarcza przykładów ani propozycji implementacji, zaś sam problem omówiono pozbawiając go jakiegokolwiek podłoża biznesowego. Pozycja ta nie wyczerpuje zatem postawionego w rozdziale 1.1 problemu.

*A multi-agent system for chemical supply chain simulation and management support* (Garcıa-Flores i Wang 2002) jest kolejną pozycją, która traktuje o modelowaniu łańcucha dostaw z wykorzystaniem MAS[[3]](#footnote-3) w oparciu o protokół komunikacyjny KQML[[4]](#footnote-4). Choć omawiany jest tutaj problem dystrybucji artykułów chemicznych, wszelkie poczynione wnioski wspólne będą dla wszystkich łańcuchów dostaw, które charakteryzują się długim czasem reakcji na bodziec oraz koniecznością produkcji pojedynczej partii o wysokim wolumenie. Zawężenie dziedziny pracy niesie ze sobą jednak korzyść w postaci silnego oparcia o konkretny proces produkcyjny (technologiczny) w firmie o określonej i rzeczywistej strukturze. Dodatkowym atutem publikacji są również przykłady implementacji pomiędzy agentami z wykorzystaniem języka KQML. Praca wpisuje się więc mocno w postawiony w pracy problem, jednak poprzestaje na zagadnieniu tworzenia symulacji. Autorzy nie wykorzystują tworzonego modelu w celu wskazania i analizy konkretnych strategii zarządzania przepływem materiałów w sieci.

*Supply chain simulation - a tool for education, enhancement and endeavor* (Holweg i Bicheno 2002) jest publikacją, która w oparciu o Lean Leap Logistics Game dowodzi tezy, jako że wykorzystanie symulacji komputerowej do analizy modelu rzeczywistego przedsiębiorstwa może przynieść realne zyski. W pracy zaprezentowany i omówiony został spadek zapasów przedsiębiorstwa, który wynikał z zsynchronizowania ze sobą poszczególnych ogniw łańcucha dostaw. Efektem analizy modelu, a także symulacji komputerowej przeprowadzonej w oparciu o niego, było także zmniejszenie buforów czasowych występujących w transporcie oraz zwiększenie stabilność popytu. Pozycja ta, o ile nie przedstawia konkretnych, uniwersalnych i skalowalnych metod zarządzania łańcuchem dostaw, dowodzi użyteczności symulacji komputerowej jako narzędzia do rozwiązywania rzeczywistych problemów biznesowych.

*Distributed Supply Chain Simulation Across Enterprise Boundaries* (Gan i inni 2000) to publikacja, która porównuje ze sobą dwa różne podejścia do tworzenia symulacji łańcucha dostaw. Pierwsze z nich zakłada integrację symulacji ze produkcyjnym środowiskiem, w którym dostępne są informacje o przepływie materiałów w sieci. Zakłada więc stworzenie wysokopoziomowej aplikacji rozszerzającej możliwości systemu SCM[[5]](#footnote-5) zaimplementowanego w firmie. Drugie podejście, bardziej teoretyczne i klasyczne, opiera się o modelowanie fizycznej sieci dystrybucyjnej przedsiębiorstwa i odwzorowaniu go jako układu dyskretnego. Następnie dokonywane są badania modelu. Oba podejścia porównywane są sobą pod względem wydajności oraz interoperacyjności. Publikacja ta jest więc dobrym źródłem informacji na temat tego jakie podejście do symulacji należy wybrać w zależności od tego, co chcemy badać. Nie dotyka jednak zagadnienia optymalizacji elementów oraz struktury badanego łańcucha dostaw. Z uwagi na czas, który minął od stworzenia artykułu należy również z ograniczoną ufnością podejść do wyników dotyczących wydajności. Stacje robocze, na których przeprowadzone były badania (Sun UltraSparc II 250MHz) dysponują mocą obliczeniową o trzy rzędy wielkości niższą od komputera wyposażonego w procesor Intel Core i7 7700K.

Popełniony powyżej przegląd literatury wskazuje jednoznacznie na to, że choć istnieje wiele pozycji wpisujących się w nurt tworzenia symulacji komputerowych łańcucha dostaw, żadna nie wyczerpuje dostatecznie problemu w kontekście poszukiwania optymalizacji. Stąd też możliwe jest wyciągnięcie wniosku, jakoby niniejsza praca nie powielała rozwiązanych już problemów. Uzasadnia to zatem jej powstanie i nadaje jej cechy rozwiązującej określony problem badawczy.

## Określenie problemu badawczego

Po dokonaniu przeglądu literatury fachowej dostrzec można, że nie istnieją pozycje, które w bezpośredni sposób odnosiłyby się do tworzenia symulacji komputerowej w celu poszukiwania możliwych optymalizacji. Żadna z wymienionych w rozdziale 1.2 publikacji nie nawiązuje też do logiki powszechnie wykorzystywanych systemów planowania opartych o MRP[[6]](#footnote-6). Istnieje zatem pewna luka, którą autor tej pracy pragnie wypełnić. Rzeczywiste przedsiębiorstwa sięgają po sprawdzone i skalowane rozwiązania takie jak – ogólnie pojęty – SAP[[7]](#footnote-7). Ponieważ sposób planowania produkcji i dystrybucji oparty o SAP rządzi się określonymi prawami oraz ograniczeniami (omówionymi dokładnie w rozdziale 2.), symulację należy stworzyć w taki sposób by poczynione w pracy wnioski mogły znaleźć zastosowanie w biznesowej rzeczywistości.

Problem badawczy dotyczy zatem skonstruowania symulacji, która działa w oparciu mechanizmy MRP i uwzględnia predefiniowane w systemach planowania mechanizmy planowania produkcji i dystrybucji. Na stworzonym modelu, zasilanym rzeczywistymi danymi, przeprowadzane będą badania weryfikujące postawione w rozdziale 1.4 tezy. Rozwiązywanym problemem będzie zatem odpowiedź na pytanie w jaki sposób planować produkcję i dystrybucję (jakie strategie przyjmować) by optymalizować kluczowe wskaźniki efektywności sieci.

## Cel pracy

Celem niniejszej pracy jest zbadanie oddziaływania określonych parametrów związanych z planowaniem produkcji i dystrybucji na funkcjonowanie łańcucha dostaw. Zamiarem autora jest dokonanie analizy wybranego łańcucha dostaw na kilku płaszczyznach:

* Zbadanie wpływu parametrów odpowiedzialnych za generowanie replenishmentów[[8]](#footnote-8) (strategii, progów) na KPI łańcucha dostaw
* Zbadanie zachowania łańcucha dostaw pod wpływem błędnego prognozowania sprzedaży oraz określenie zakresu możliwości reagowania na nieoczekiwany popyt
* Zbadanie wpływu zdarzeń losowych takich jak kradzież towaru, uszkodzenia linii produkcyjnych czy rozbieżności pomiędzy danymi w systemach SCM a rzeczywistością na funkcjonowanie łańcucha dostaw

Analiza wybranych łańcuchów dostaw przeprowadzona zostanie w oparciu o rzeczywiste dane i posłuży do wyciągnięcia zarówno generycznych jak i odnoszących się do konkretnego przypadku wniosków, jakie strategie zarządzania łańcuchem dostaw należy przyjąć, by parametry sieci zbliżyły się do optimum.

Z uwagi na mnogość scenariuszy możliwych do zbadania za pomocą symulacji oraz chęć rozszerzenia zakresu badań poza same parametry systemów planowania w pracy zweryfikowane zostaną następujące hipotezy:

|  |
| --- |
| Możliwe jest dokonanie poprawy funkcjonowania łańcucha dostaw poprzez odpowiednie skalibrowanie parametrów systemów planowania. |

|  |
| --- |
| Niska dokładność prognoz sprzedaży wprowadza do łańcucha dostaw istotne zaburzenia znacząco wpływające na KPI. |

|  |
| --- |
| Symulacja komputerowa jest efektywnym sposobem analizowania łańcucha dostaw i umożliwia optymalizację rzeczywistych systemów |

# Planowanie produkcji i dystrybucji w środowisku SAP

## Wstęp – model MRP

Planowanie produkcji i dystrybucji w środowisku SAP opiera się o model MRP. Istnieje wiele standardów oraz implementacji tego modelu, jednak wspólne dla wszystkich ramy znajdują się na rysunku 1.



Rysunek - Model MRP, Źródło: Opracowanie własne

Model MRP opiera się na logice, w której pierwotnie określany jest generalny, ramowy plan funkcjonowania całego przedsiębiorstwa, a następnie, wraz z każdym krokiem, jest on uszczegóławiany aż do poziomu egzekucji produkcji poszczególnych dóbr. Występują w nim sprzężenia zwrotne, które zakładają, że plan idealny może być niemożliwy do osiągnięcia. W takim przypadku należy stworzyć plan, który możliwy będzie do realizacji po uwzględnieniu dostępnych zasobów – materiałowych, produkcyjnych, ludzkich. To właśnie ten element (RCCP[[9]](#footnote-9)) stanowi główną różnicę pomiędzy modelami MRP oraz MRP II.

Pierwszym krokiem w MRP jest planowanie wysokiego poziomu. To tutaj określane są cele biznesowe organizacji – strategia rozwoju, portfolio produktów, sposoby docierania do potencjalnych klientów, a także sposób pozyskiwania sprzedawanych dóbr. Wszystkie plany są generyczne i na tym etapie nie są uwzględniane żadne rzeczywiste ograniczania. Momentem przejścia na niższy poziom jest tworzenie planu strategicznego. Jest to ciągle planowanie na wysokim poziomie, jednak następuje tutaj rozbicie produktów na pewne rodziny dla których zakładane są określone wolumeny sprzedaży. To właśnie względem tych wolumenów tworzona będzie prognoza finansowa przychodów przedsiębiorstwa. Zakładane wielkości sprzedaży stanowią również bazę do kalkulacji wymaganych zasobów produkcyjnych.

Jeżeli analiza wysokiego poziomu wskazuje na to, że zakładane cele są możliwe do osiągnięcia (to znaczny – nie istnieją teoretyczne ograniczenia produkcji i dystrybucji, które mogłyby w oczywisty sposób uniemożliwić osiągnięcie zadanego poziomu sprzedaży) następuje przejście na poziom planowania szczegółowego. Właśnie na tym etapie model MRP dostarcza odpowiedzi na pytanie, co, kiedy i w jakiej ilości produkować/zamawiać. Tworzony jest więc plan replenishmentu idealnego. Jednocześnie na tym etapie występować będzie najwięcej ograniczeń, które wymagać będą dostosowania rzeczywistego planu produkcji/dostaw do teoretycznie idealnych zapotrzebowań materiałowych. Wraz ze wzrostem różnorodności oferowanych produktów oraz zaawansowaniem technologii ich wytwarzania, uwidacznia się pewna luka pomiędzy prostym modelem MRP a rzeczywistością. Ilość danych i potencjalnych scenariuszy do przeanalizowania zdecydowanie przerasta możliwości człowieka. Jednocześnie sam model MRP nie dostarcza odpowiedzi na pytanie jak powinno przebiegać RCCP. Z pomocą przychodzą jednak rozszerzenia SAP omówione dokładnie w rozdziale 2.2.2.

Planowanie szczegółowe nie ogranicza się wyłącznie do ilości, kolejności i momentu produkcji wyrobów gotowych. Należy zdać sobie sprawę z występowania wzajemnej zależności komponentów, półproduktów oraz produktów końcowych. Wielokroć to właśnie one są głównym ograniczeniem i jednoznacznie decydują o możliwości, bądź braku możliwości wykonania produkcji. Oczywistym sposobem przeciwdziałania wahaniom zapotrzebowań i dostaw jest wprowadzenie buforów, odpowiednio zdefiniowanych progów replenishmentu. Stanowią one często nie tylko dodatkowe zabezpieczanie, ale też kompensację rzeczywistych ograniczeń związanych z produkcją/dostawami.

Upewniwszy się, że stworzony plan produkcji jest odpowiedzią na realne zapotrzebowania i możliwy jest do wykonania z punktu widzenia dostępnych zasobów, konieczne jest przejście do trzeciego i ostatniego głównego bloku MRP – egzekucji i kontroli. Głównym celem jest weryfikacja tego, czy stworzony wcześniej (zgodny z teoretycznymi założeniami) plan wykonywany jest zgodnie z harmonogramem. Być może założona wydajność linii produkcyjnej czy stacji roboczej nie odpowiada rzeczywistości i wymaga korekty przy kolejnych kalkulacjach. Może zdarzyć się tak, że procent odpadów z produkcji przekracza założony próg i by wyprodukować wymaganą ilość pełnowartościowego produktu należy tak naprawdę brać pod uwagę zwiększone zapotrzebowanie na komponenty. Ostatecznie – teoretyczne zużycie materiałów obliczone przez inżyniera procesu może różnić się od rzeczywistego, co wymaga korekcji BOM[[10]](#footnote-10). Potencjalnych scenariuszy może być wiele, jednak każde odstępstwo od teoretycznych założeń powinno zostać odnotowane, a sam model skorygowany w taki sposób, by następna kalkulacja zaowocowała dokładniejszymi i bliższymi prawdy obliczeniami.

Jak można się domyślić – cały proces MRP jest dynamiczny. Z uwagi na ilość czynników wpływających na określony efekt końcowy kalkulacje zapotrzebowań prowadzone są z co najmniej dobową częstotliwością. Możliwe byłoby oczywiście doprowadzenie do sytuacji, w której kalkulacja prowadzona byłaby w czasie zbliżonym do rzeczywistego, jednak wbrew pozorom nie jest to stan w pełni pożądany. Owocowałoby to brakiem spójności obrazu widzianego z różnych perspektyw – planowania popytu, planowania dystrybucji, planowania produkcji, planowania materiałowego. Dodatkowo należy wziąć pod uwagę tzw. efekt byczego bicza – małe zmiany na początku łańcucha zależności owocują bardzo dużymi wahaniami na jego końcu.

Dlatego właśnie większość rzeczywistych przedsiębiorstw definiuje interwały odświeżania składowych modelu MRP. W przypadku omawianej w kolejnych rozdziałach firmy X są one następujące:

* Plan strategiczny – 1 miesiąc
* Prognoza popytu – 1 tydzień
* Długoterminowy plan produkcji – 1 tydzień
* Krótkoterminowy plan produkcji – 1 dzień

## Implementacja modelu MRP w SAP

Opisany w rozdziale 2.1 model MRP można w rzeczywistości w stosunkowo prosty sposób przekształcić do postaci, która umożliwi precyzyjne obliczanie zapotrzebowania materiałowego. Aby to uczynić należy określić przestrzeń elementów MRP oraz ich dodatni/ujemny wpływ na bilans kalkulacji. Istnieje tutaj pełna dowolność w kwestii notacji i jest to zależne od konkretnej implementacji SAP. W przypadku badanej firmy X wyróżnić można następujące elementy MRP:

Tabela - Wykaz dodatnich elementów MRP

|  |  |  |
| --- | --- | --- |
| **Element** | **Pełna nazwa** | **Opis elementu** |
| Stock | Stock | Zapas, fizycznie dostępny produkt spełniający określone wymogi jakościowe, który można zaoferować klientowi, bądź wykorzystać do produkcji wyższego rzędu |
| PchOrd | Purchase Order | Potwierdzona dostawa produktu (pozostająca w fazie planowania), która fizycznie nie opuściła ogniwa zasilającego |
| ShipNt | Shipment Notification | Potwierdzona dostawa produktu, która fizycznie opuściła ogniwo zasilające i aktualnie znajduje się w tranzycie |
| PlOrd | Planned Order | Zależnie od ogniwa – niepotwierdzona produkcja bądź zlecenie replenishmentu |
| PrcOrd | Process Order | Potwierdzona produkcja towaru, która zlecona została fabryce |
| QMLot | Quality Inspection Lot | Wyprodukowany towar, który nie przeszedł kontroli jakości, nie można zaoferować go klientowi, bądź wykorzystać do produkcji wyższego rzędu |

Tabela - Wykaz ujemnych elementów MRP

|  |  |  |
| --- | --- | --- |
| **Element** | **Pełna nazwa** | **Opis elementu** |
| Safety Stock | Safety Stock | Próg replenishmentu – statyczna wartość, określająca docelową wartość bilansu w danym ogniwie łańcucha |
| Order | Order | Niepotwierdzone zamówienie klienta, może ono nie mieć pokrycie w fizycznie dostępnym zapasie |
| Deliv | Delivery | Potwierdzone zamówienie klienta mające fizyczne pokrycie w zapasie, bądź zlecenie replenishmentu to ogniwa niższego rzędu |
| DepReq | Dependent Requirement | Zapotrzebowanie zależne, wykorzystanie w produkcji wyższego rzędu |
| PlORel | Planned Order Release | Otrzymanie replenishmentu od ogniwa niższego rzędu |
| IndReq | Independent Requirement | Prognoza sprzedaży |

Wymienione wyżej elementy tworzyły będą tak zwane MRP listy – czyli zestawienia odpowiednich ruchów materiałowych, które umożliwiają dokonania kalkulacji zapotrzebowania, tak by zapewniona była ciągłość i terminowość dostaw towarów.

Przeanalizujmy kilka przykładów MRP List:



Rysunek - MRP Lista produktu 1

Jest to wyjątkowo prosty przypadek. Obecny zapas produktu wynosi 147 [CS]. Każdego dnia spodziewana jest sprzedaż na poziomie 6-9 [CS] reprezentowana przez elementy Independent Requirement. Obecnie żaden z klientów nie zdecydował się na założenie zamówienia – nie są spodziewane są więc żadne wysyłki – stąd brak elementów typu Order czy Delivery. Nie są spodziewane również żadne zasilenia ogniw niższego rzędu, co sugeruje brak elementów PlORel. Możliwe do zauważenia jest natomiast zasilenie z lokacji 5053. Co istotne - występuje przed momentem, gdy projekcja stanu magazynowego osiągnie poziom 0. Można więc określić próg repelnishmentu dla tego produktu. Wynosi on około 50 [CS]. Zwrócić uwagę należy również na wielkość samego zasilenia, która wynosi 100 [CS]. Nie jest to oczywiście wartość przypadkowa. W logice SAP dla każdego produktu określana jest rozdzielczość replenishmentu – minimalna ilość produktu, która może być dostarczona z ogniwa wyższego rzędu. W tym przypadku wynosi ona 100 [CS]. Nie wynika ona z żadnej kalkulacji, a jest odzwierciedleniem biznesowych realiów. W tym konkretnym przypadku do lokacji A751 możemy dostarczyć minimum jedną paletę, a dla produktu 83725610 oznacza to równowartość 100 kartonów [CS].



Rysunek - MRP Lista produktu 2

Również i w tym przypadku mamy do czynienia ze stosunkowo mało skomplikowaną sytuacją. W pierwszej kolejności uwagę należy zwrócić na obecność elementów Quality Inspection Lot oraz Process Order. Sugerują one od razu, że lokacja 2725 jest w rzeczywistości fabryką. Ponieważ oba te elementy występują jednocześnie (tzn. z tą samą datą) wyciągnąć można wniosek, że zlecona została produkcja 1452 [CS] produktu. W chwili generowania listy wyprodukowano już 396 [CS], ciągle odbywa się produkcja pozostałych 1056 [CS]. Wyprodukowane 396 [CS] nie są jednak możliwe do wykorzystania do wysyłki lub sprzedaży, ponieważ nie została przeprowadzona kontrola jakości produktów – nie są one jeszcze zapasem. 10.08 zaplanowana jest kolejna produkcja. Ponieważ nie została jeszcze ona zlecona, a jedynie wstępnie zaplanowana, reprezentowana jest przez MRP element, którym jest Planned Order.

Omówioną uprzednio produkcję zlecono ponieważ do fabryki spływają zlecenia replenishmentów z ogniwa niższego rzędu – lokacji 2621. Jest to jednocześnie jedyna lokacja, która wymaga zasilenia.

Należy również zwrócić uwagę na projekcje stanu magazynowego. Przed produkcją 10.08 wynosi ona aż -3036 [CS]. Potrzeby zasilenia ogniwa niższego rzędu znacznie przewyższają więc zaplanowaną produkcję. Konieczna jest zatem taka korekta planu produkcyjnego, by zaspokoić wszystkie zapotrzebowania.



Rysunek - MRP lista produktu 3

Jest to przypadek zdecydowanie bardziej skomplikowany niż dwa poprzednie. W momencie generowania MRP listy do dyspozycji mamy 347 [CS]. Kolejne 324 [CS] znajdują się w tranzycie – odzwierciedlone jest to za pomocą elementu Shipment Notification. Ogniwem zasilającym jest 2621.

Zarówno obecnie dostępny zapas jak, produkty w tranzycie, jak i przyszłe replenishmenty wykorzystywane zostaną do produkcji wyższego rzędu. Zdradza to obecność zapotrzebowania zależnego. Część z tej produkcji została już zlecona (potwierdzone zapotrzebowania zależne – Order Reservation), część produkcji pozostają nadal w fazie planowania (Dependent Requirement).

Aby zaspokoić przyszłe zapotrzebowania zależne występujące w lokacji 2727 konieczne jest uzupełnienie zapasu – stąd wygenerowane są kolejne zasilenia z lokacji 2621. Jak widać – w żadnym momencie projekcja stanu magazynowego nie jest mniejsza niż 0. Oznacza to, występuje równowaga pomiędzy występującymi zapotrzebowaniami i zasileniami.

Nie jest również prognozowana sprzedaż produktu w lokacji 2727 – nie występują żadne zapotrzebownia niezależne (Independent Requirement). Z tego powodu nie jest konieczne definiowanie progu replenishmentu buforującego wahania zapotrzebownia i dostaw.

## Rozszerzenia SAP udoskonalające planowanie MRP

### Ograniczenia logiki MRP

W rozdziale 2.1 omówione zostało dokładnie działanie modeli MRP oraz MRP II. Różnica pomiędzy nimi polegała głównie na RCCP, czyli uwzględnianiu ograniczonych możliwości produkcyjnych oraz dystrybucyjnych przy tworzeniu planu replenishmentów. Niestety sam proces RCCP, zasady wybierania priorytetów a także metody i algorytmy balansowania dostaw nie zostały w żaden sposób określone w MRP. Z pomocą nie przychodzi również sama implementacja MRP w SAP. Wszystkie procesy podejmowania decyzji przeniesione zostały na użytkowników systemu. Jest to luka, która uwydatnia się w szczególności w przypadkach bardzo ograniczonych możliwości dostaw oraz względnie wysokiego zapotrzebowania – kiedy stosunek . Sytuacje te wymagają bardzo dużej ilości ręcznych interwencji w plan dostaw generowany przez MRP, co znacząco pomniejsza zyski płynące z wdrożenia SAP.

Kolejnym bardzo istotnym ograniczaniem MRP jest brak możliwości zróżnicowania klas zapotrzebowania. Wszystkie elementy MRP mają dokładnie taką samą wagę i taki sam priorytet. Nie oddaje to w żaden sposób biznesowych realiów, gdzie np. zamówienie klienta nieznajdujące pokrycia w zapasie będzie zdecydowanie ważniejsze niż zapotrzebowanie zależne produktu, czy zapotrzebowania dystrybucji z ogniw niższego rzędu. W szczególności , gdy replenishmenty potrzebne są jedynie, by osiągnąć parametry bezpieczeństwa. Nastręcza to problemów zarówno podczas planowania produkcji, jak i dystrybucji (kiedy dzielimy produkty pomiędzy ogniwami niższego rzędu, gdzie źródła powstawania zapotrzebowani mogą być znacząco różne).

Trzecim głównym ograniczeniem logiki MRP jest brak możliwości kontrolowania przepływu materiałów w sieci. Jest to problem, który dotyczyć będzie przede wszystkim sieci i złożonej strukturze, w której występują wielopoziomowe replenishmenty. W takich przypadkach MRP nie posiada informacji na temat tego, w których ogniwach sieci najkorzystniej jest posiadać zapas. Kryteria wyboru mogą być różne (np. faktyczny koszt magazynowania towarów, czy też znacząco różniące się prognozy sprzedaży w poszczególnych ogniwach sieci), jednak niezależnie od podłoża, logika MRP nie posiada mechanizmu wspierającego podejmowanie takich decyzji.

Wymienione w tym rozdziale ograniczania nie wyczerpują rzecz jasna problemu niedoskonałości MRP. Potencjalne problemy można by zapewne znaleźć analizując konkretne problemy biznesowe. Przytoczone jednak luki wspólne są dla wszystkich systemów opartych o MRP niezależnie od obecnych tam procesów, czy rodzaju prowadzonego biznesu.

Można byłoby w tym momencie poczynić wniosek, jakoby rozważane w tym rozdziale ograniczenia eliminowały MRP (oraz jego implementację w SAP) jako skuteczną metodę planowania nowoczesnego przedsiębiorstwa produkcyjno-dystrybucyjnego. Należy jednak powstrzymać się od zbyt szybkiej oceny i zdać sobie sprawę, że wdrożenie jakiegokolwiek oprogramowania klasy ERP[[11]](#footnote-12) wymaga bardzo dużych nakładów finansowych i czasowych. Częstokroć implementacja tego typu rozwiązań nieodzownie związana jest z modyfikacją obecnie istniejących procesów, ich eliminacją, czy też zastąpieniem nowymi procedurami. Podobnie sam proces integracji zaburza efektywność działania przedsiębiorstwa. Dlatego właśnie dużo bardziej rozsądnym jest uzupełnienie istniejącego już rozwiązania o rozpoznane braki, niż tworzenie oraz implementacja zupełnie nowego oprogramowania.

W rozdziałach 2.3.2 oraz 2.3.3 omówione zostaną 2 rozwiązania, które eliminują omówione wcześniej ograniczenia. Mogą to być rozwiązania dostarczone bezpośrednio przez SAP, dostosowane do potrzeb konkretnego przedsiębiorstwa, jak i produkty zewnętrznych dostawców zintegrowane z SAPem.

### APS

Główne ograniczenia MRP w kontekście planowania produkcji dotyczyły braku algorytmów balansowania oraz nadawania priorytetów produkcji, a także konieczność rozpatrywania zbiorczych zaporzebowań w sieci zamiast aktualnego stanu każdego z węzłów. Odpowiedź na ten problem stanowi oprogramowanie klasy APS[[12]](#footnote-13). Jest ono całkowicie zintegrowane z SAP. Za pomocą interfejsu pobiera dane na temat zapasów oraz zapotrzebowań. Posiada zdefiniowane koszty produkcji. Dysponuje również odpowiednio skonstruowanym systemem kar – za całkowite wyczerpanie zapasu, projekcję zapasu poniżej progu replenishmentu, a także projekcję zapasu powyżej maksymalnego dopuszczalnego zapasu. Działanie systemu APS opiera się więc na programowaniu liniowym oraz wielowymiarowym minimalizowaniu kar.

Uzupełnieniem warstwy algorytmicznej jest możliwość implementacji rozbudowanego graficznego interfejsu użytkownika, co niemożliwe jest w natywnym środowisku SAP. Wbrew pozorom GUI nie stanowi jedynie estetycznego uatrakcyjnia aplikacji, ale może znacząco rozszerzać funkcjonalność np. poprzez wyświetlenie BOM dla każdej z zaplanowanych produkcji po jej podświetleniu, zmianę harmonogramu produkcji za pomocą metody przeciągnij i upuść, czy informowaniu o potencjalnych brakach materiałowych za pomocą komunikatów pop-up.

Główne korzyści płynące z wdrożenia sytemu APS:

1. **DC Network** - możliwość planowania produkcji uwzględniając zapotrzebowania każdego ogniwa w sieci osobno, nie zaś rozważając zbiorcze zapotrzebowania występujące w ogniwie zasilającym.
2. **Minimum Cost Flow** - możliwość przyznawania pierwszeństwa wybranym ogniwom w sieci na podstawie zróżnicowanego systemu kar za wykraczanie poza pożądane okno zapasu.
3. Balansowanie produkcji na poziomie dnia/tygodnia/miesiąca, możliwość planowania produkcji z wyprzedzeniem, kiedy spodziewane są problemy z dostępnością zasobów produkcyjnych w przyszłości (tzw. prebuild).
4. **GUI -** dynamiczna informacja o brakujących komponentach, zasobach, wykorzystaniu maszyn.
5. Możliwość przeprowadzania analiz what-if w celu oceniania różnych potencjalnych scenariuszy planowania produkcji.

### APO

Moduł APO[[13]](#footnote-14) odpowiada za udoskonalone planowanie dystrybucji. Jego główną różnicą względem planowania MRP jest tworzenie dwóch równoległych planów replenishmetnów generowanych za pomocą dwóch silników:

1. **Optimizer** – tworzy plan w krótkim horyzoncie (6 tygodni), oparty jest na ograniczonych zasobach i reguluje przepływ materiałów w sieci poprzez minimalizację kosztów: magazynowania produktu, niedostępności produktu w danym ogniwie, opóźnienia replenishmentu.
2. **Heuristic** – tworzy plan w długim horyzoncie (18 miesięcy), oparty jest on na nieograniczonych zasobach i odzwierciedla idealny plan replenishmentu. Jest więc on dokładnym odzwierciedleniem planowania materiałowego MRP.

Plan generowany przez Optimizera jest więc wykorzystywany podczas planowania wysyłek, zasileń ogniw niższego rzędu, zaś plan stworzony przez Heuristic będzie stanowić bazę do planowania produkcji. APO, podobnie jak APS wymaga zdefiniowania kosztów (kar) za magazynowanie produktu w danym węźle sieci, a także wykraczania poza zdefiniowane okno zapasu. Tworzenie planu dystrybucji polega więc na minimalizowania kosztów w sieci.

Główne korzyści płynące z wdrożenia APO:

1. **Invetory Assignment Tool** – narzędzie agregujące wszystkie informacje niezbędne do podjęcia decyzji o zaplanowaniu transferu materiałów pomiędzy węzłami sieci: nieograniczony plan dystrybucji, ilość zamówień i prognoza sprzedaży w węźle do którego planowana jest dostawa, ilość produktu w całej sieci, zaplanowane kolejne produkcje/dostawy, priorytet dostarczenia produktu oraz – co najważniejsze – sugerowana wielość oraz data wysyłki.
2. **Coverage Report** – projekcja stanu magazynowanego produktu we wszystkich ogniwach w sieci uwzględniająca ograniczone zasoby produkcji/dystrybucji, która pozwala wychwycić oraz zmitygować potencjalne ryzyko niedostępności produktów.
3. **Planning Book** – podsumowanie zamówień, zapotrzebowań, replenishmentów które pozwala dokonywać analiz projekcji stanu magazynowego oraz analiz what-if w zdefiniowanych agregacjach czasowych: dobowych, tygodniowych, miesięcznych.
4. **Interchangeability** – narzędzie umożliwiające łącznie par produktów PIPO[[14]](#footnote-15), by możliwe było planowanie pary produktów w taki sam sposób, jak gdyby były one jednym produktem. Minimalizuje to ilość pozostałości na produkcie, który wycofywany jest ze sprzedaży i umożliwia opóźnienie produkcji nowego produktu, co przekłada się na mniejszą ilość zapasu w sieci, bez stwarzania ryzyka niedostępności produktu.
5. **Time Phased Safety** – funkcjonalność umożliwiająca tworzenie profili bezpieczeństwa zależnych od czasu. Nie stanowią one w takim wypadku integralnej części tzw. danych stałych i umożliwiają stosowanie różnych strategii bezpieczeństwa (np. podniesienie progu replenishmentu o 20% podczas okresu wakacyjnego), zależnych od bieżących potrzeb.
6. **Integracja z Business Warehouse** (SAP BO) – archiwizacja i zapewnienie dostępu do kalkulacji przeprowadzonych w przeszłości, tak by możliwe było przeprowadzanie analiz z ich wykorzystaniem.

## Powiązania pomiędzy MRP, SAP a stworzoną symulacją komputerową

Celem niniejszej pracy nie jest wyłącznie odwzorowanie samego modelu MRP, jego ulepszenie, czy też integracja MRP z modułami klasy APO/APS. W najlepszym przypadku implementacja taka zaowocowałaby stworzeniem kopii opisanych w rozdziale 2. rozwiązań To w oczywisty sposób nie umożliwiłoby weryfikacji postawionych w rozdziale 1.4. tez. Dlatego właśnie stworzona na potrzeby tej pracy symulacja komputerowa (choć oparta o MRP i dziedzicząca po SAP logię kalkulacji) służyć będzie przede wszystkim badaniu zjawisk i zależności w łańcuchu dostaw.

Model MRP będzie więc jedynie punktem wyjściowym dla szerszego i bardziej skomplikowanego problemu, jakim jest efektywne zarządzanie łańcuchem dostaw. Dużo łatwiej powielić dobre praktyki, które powstały w oparciu o istniejące i sprawdzone rozwiązania, niż te, które wypracowane zostały wyłącznie na podstawie teoretycznych rozważań niekoniecznie znajdujących odzwierciedlenie w rzeczywistości. Jeżeli zaś sam model zasilony zostanie rzeczywistymi danymi dotyczącymi prognozowanej sprzedaży i faktyczną sprzedażą możliwe będzie wyciągnięcie znaczących oraz wiarygodnych wniosków.

Co pozostanie jednak bez zmian to sama logika kalkulacji MRP wraz z jej ograniczeniami, nomenklatura samych elementów MRP omówionych w rozdziale 2.2., a także sposób przeglądania wyników kalkulacji zgodny z konwencją MRP Listy w SAP.

Możliwe byłoby w tym momencie dokonanie pewnej krytyki implementowanego rozwiązania. Jeżeli powszechnie dostępne są pewne komercyjne rozwiązania umożliwiające minimalizację lub wręcz eliminację niedoskonałości modelu MRP, jaki sens ma przeprowadzanie kalkulacji i wyciąganie wniosków w oparciu o symulację komputerową pozbawioną cech systemów APO lub APS? Jest to pytanie co najmniej słuszne, lecz odpowiedź na nie jest prosta. Systemy klasy APO/APS rozszerzają możliwości MRP, lecz nie zastępują go zupełnie. Istnieje ponadto wiele procesów, których rozwiązania te nie są w stanie pokryć. Najlepszym przykładem może być długoterminowe planowanie materiałowe, które w wielu rzeczywistych przedsiębiorstwach ciągle przeprowadzane jest ręcznie, tylko i wyłącznie na podstawie list MRP. Tak więc możliwe staje się wyciągnięcie dosyć prostego wniosku, opartego na powszechnej w informatyce zasadzie GI-GO[[15]](#footnote-16) - jeżeli systemy rozszerzające funkcjonalności MRP zasilane są danymi niskiej jakości, również i one generować będą nie znaczące tak wiele rezultaty. Co jest również istotne – wiele generycznych wniosków dotyczących zarządzania łańcuchem dostaw poczynionych w wyniku rzeczonej symulacji będzie mogło znaleźć zastosowanie w przypadku przedsiębiorstw wykorzystujących APO/APS.

# Modelowanie łańcucha dostaw

## Model odzwierciedlony w symulacji

Zaimplementowany w symulacji model odpowiada sieci dystrybucyjnej przedsiębiorstwa X. Składa się z 9 ogniw, które reprezentują fizyczne miejsca składowania dóbr. Przepływ dóbr w sieci pomiędzy parą dwóch dowolnych ogniw jest wyłącznie jednokierunkowy. Wyłącznie jedno ogniwo w całej sieci to tzw. customer facing DC[[16]](#footnote-17) (Rysunek 5, sygnatura 2621). Produkty wytwarzane są w czterech fabrykach (Rysunek 5, sygnatury 5053, 4850, 4853, 2725). Wszystkie dobra importowane poddawane są kontroli celnej i składowane są w magazynie celnym (Rysunek 5, sygnatura 2751). Dodatkowo zasilenia z niektórych fabryk nie odbywają się bezpośrednio, lecz z przeładunkiem w pośrednim centrum dystrybucyjnym (Rysunek 5, sygnatura 9979). Ostatnim elementem w sieci są dwa centra manipulacji (Rysunek 5, sygnatury 2727 oraz 2742), w których świadczone są VAS[[17]](#footnote-18). Do nich trafiają gotowe produkty, które podlegają manipulacji i wracają do głównego centrum dystrybucyjnego już jako inne dobra.

Na opisanym w tym rozdziale modelu prowadzone będą wszystkie badania, jednak nie oznacza to w żadnym wypadku, że stworzona aplikacja posiada ten model twardo zapisany w kodzie. Wręcz przeciwnie – wszystko konfigurowalne jest z poziomu użytkownika, a zaimplementowane algorytmy gwarantują poprawność obliczeń dla dowolnego modelu sieci.



Rysunek - Struktura badanego łańcucha dostaw

### Elementy modelu

W omawianym modelu można wyróżnić cztery główne elementy:

1. Lokacje (*LOCATIONS*)
2. Produkty (*PRODUCTS*)
3. Ścieżki transportowe (*TLANES*)
4. Parametry bezpieczeństwa (*SAFETIES*)

Zbiór lokacji określa ogniwa łańcucha dostaw. Jest to generyczna kategoria opisująca wszystkie węzły w sieci dystrybucyjnej, niezależnie od ich kategorii. Każda lokacja posiada unikalną sygnaturę (*plantcode*), która jest jej wyróżnikiem. Aby lokacje były łatwo rozróżnialne dla użytkownika, do każdej lokacji można przypisać również opis (*desceription*), w którym zawarte mogę być dodatkowe informacje takie jak np. nazwa fabryki, czy centrum dystrybucyjnego. Konieczne jest również określenie typu lokacji (*type*), który jednoznacznie odpowiada na pytanie, czy dana lokacja jest węzłem wytwarzającym dobra (typu *P*), czy węzłem przesyłającym je dalej, lub miejscem ich konsumpcji (w obu przypadkach – typ *F*).

Zbiór produktów określa przestrzeń dóbr przepływających przez sieć dystrybucyjną. Każdy z nich posiada unikalną sygnaturę (*gcas*), który pozwala na jednoznaczną identyfikację produktu podczas wszystkich przepływów informacji występujących wewnątrz przedsiębiorstwa. Nie musi on jednocześnie być wykorzystywany przez klientów, którzy znacznie częściej opierają się na kodach kreskowych (EAN[[18]](#footnote-19)). Każdy produkt posiada również opis (*description*), który ułatwia jego identyfikację. Może on np. zawierać nazwę handlową dobra, rozmiar produktu, ilość opakować w jednostce itd. Są to informacje zupełnie zbędne z punktu widzenia planowania zapotrzebowania materiałowego, jednak bardzo przydatne z punktu widzenia operatora systemu. Kolejną istotną cechą produktu jest jego jednostka miary (*uom*), która określa jak mierzmy ilość produktu (czy jest to kilogram, sztuka, paleta itp.). Konieczne jest również określenie typu dobra (*type*), by możliwe było zidentyfikowanie, czy jest to wyrób gotowy (*FERT*), półprodukt (*HALB*), materiał surowy (*RAW*), czy też opakowanie (*PACK*). Determinuje to również pozycję w hierarchii wykonywanych obliczeń. Najpierw kalkulowane są zapotrzebowania niezależne (czyli produktów końcowych), a dopiero później zależne (czyli pochodne planowanej produkcji i dystrybucji). Ostatnią rozpatrywaną cechą produktu jest jego wielkość zaokrąglania (*roundval*). Określa jednocześnie minimalną ilość produktu, którą można przesłać pomiędzy ogniwami w sieci, a także wielkość replenishmentu (który jest zawsze jej wielokrotnością). Konieczność jej stosowania wynika z fizycznych i biznesowych ograniczeń – np. braku możliwości przewiezienia ilości mniejszej niż jedna paleta.

Kolejnym elementem modelu są ścieżki transportowe. Określają one relacje pomiędzy poszczególnymi ogniwami w sieci – ogniwo początkowe (*startloc*) oraz ogniwo docelowe (*endloc*). Zwarte są tutaj również informacje na temat czasu trwania transportu pomiędzy ogniwami (*duration*) oraz fizycznej odległości pomiędzy nimi (*distance*).

Ostatnim, ale niekoniecznie najmniej ważnym, elementem zaimplementowanego modelu są parametry bezpieczeństwa. Rozumieć je należy jako parametry definiujące progi replenishmentu dla każdej pary (*product, location*). Zasadniczo rozróżnić możemy dwie strategie generowania zasileń. Statyczną, na podstawie stałej, zadanej wielkości oraz dynamiczną, która gwarantuje pokrycie na *n* jednostek czasu, niezależnie od wielkości zapotrzebowań występujących w tak określonym oknie czasowym. W stworzonym modelu znajduje to za odzwierciedlenie w dwóch parametrach. Strategia (*strategy*), informacje nas w jaki sposób generowane będą zasilenia, a pokrycie (*quantity*) odpowiada na pytanie poniżej jakiego progu wymagany będzie replenishment, albo na jak długo w przyszłość chcemy zapewnić pokrycie materiałowe.

### Założenia upraszczające model

Pomimo, że każda tworzona symulacja komputerowa z natury zakładać powinna implementację modelu pozwalającego na uzyskanie jak najdokładniejszych wyników, niektóre zjawiska istniejące w rzeczywistości są trudne (czy też niemożliwe) do odtworzenia podczas procesu modelowania.

Jednym z najpoważniejszych problemów występujących w łańcuchach dostaw jest wspominany w rozdziale 2.1 efekt byczego bicza – nawet małe zaburzenia występujące na początku łańcucha dostaw mogą doprowadzić do bardzo dużych zaburzeń na jego końcu. Źródeł takich zaburzeń może być wiele – począwszy od niedokładnej prognozy sprzedaży, poprzez opóźnienia w załadunku/rozładunku pojazdów, uszkodzenie towaru w magazynie, katastrofę naturalną, spóźnienie się auta na prom, skończywszy na promocji dla konsumentów, która nie była komunikowana przez klientów i idące za tym zwiększone zapotrzebowanie materiałowe.

Choć z technicznego punktu widzenia możliwe jest zasymulowanie dowolnego z tych zjawisk, sporne jest czynienie tego bez określonego kontekstu biznesowego. Dlatego w stworzonym modelu rozpatrywane nie będą elementarne źródła takich zakłóceń. Zebrane one zostaną w grupę np. opóźnionej dostawy towaru, albo niedostępności towaru pomimo wcześniejszej awizacji. Dla tak generycznie określonych problemów badanie będzie zachowanie się modelu. Podobnie również wrażliwość modelu badana będzie dla zgeneralizowanych czynników.

Kolejnym uproszczeniem jest rozdzielczość czasu. W rzeczywistych przypadkach częstokroć godziny decydują o powodzeniu lub braku powodzenia określonego przedsięwzięcia, wpływając jednocześnie na wskaźniki efektywności. Co więcej – istnieją dedykowane systemy takie jak TMS [[19]](#footnote-20)lub YMS[[20]](#footnote-21), zastosowanie których pozwala z bardzo wysoką precyzją zarządzać poszczególnymi slotami rozładunkowymi/załadunkowymi i planować wykorzystanie infrastruktury magazynowej w oparciu o awizacje. Również i w takim przypadku występować mogą wspomniane wcześniej zaburzenia, które w istotny sposób wpływają na cały łańcuch dostaw. Nietrudno wyobrazić sobie, wcale nierzadką, sytuację, w której magazyn pracuje wyłącznie w trybie dwuzmianowym, a spóźniony transport dociera do magazynu w piątek, poza godzinami pracy, gdzie kierowca będzie zmuszony oczekiwać na rozładunek przez 2 dni. Przełożenie tej sytuacji na dostępność towaru i możliwość realizacji zamówień jest oczywiste. Sytuacje jednak te nie zostały jednak dokładnie odwzorowane w symulacji. Minimalną jednostką czasu, na jakiej operuje symulacja jest pełna godzina. Wszystkie awizacje, procesy produkcyjne, zwolnienia z kontroli jakości itp. są dokładnie opisane przez czas, kiedy występują. Nie jest jednak dokonywana analiza tego czasu na poziomie niższym niż godzinowy. Nie jest to wymagane z perspektywy potencjalnego czasu pojedynczego przebiegu symulacji, który trwać może kilka miesięcy.

Poczynione uproszczenia wynikają z tego, że przedmiotem tej pracy nie pozostaje optymalizacja wszystkich procesów w łańcuchu dostaw, ani też zasymulowanie wszystkich możliwych rzeczywistych przypadków. Zgodnie z postawionymi w rozdziale 1.4 tezami badany jest wpływ parametrów odpowiedzialnych za planowanie na wybrana wskaźniki efektywności. Oznacza to, że w zakresie rozważań pozostaje odpowiedzenia na pytanie takie jak ‘w którym ogniwie łańcucha dostaw optymalne jest posiadanie zapasu?’, ‘jakie parametry bezpieczeństwa należy utrzymywać, by uzyskać pożądany poziom obsługi klienta?’, ‘jak rozmiar minimalnego wolumenu produkcyjnego wpływa na ilość towaru w łańcuchu dostaw?’, bądź też ‘jakie przełożenie na funkcjonowanie łańcucha dostaw ma prognoza sprzedaży o określonej wartości MAPE[[21]](#footnote-22)?’. Aby znaleźć odpowiedzi na te pytania konieczne jest przeprowadzenie generycznej i wysokopziomowej analizy łańcucha dostaw. Z uwagi na czas tranzytu pomiędzy ogniwami w badanym modelu, który wynosi do kilkunastu dni, często by zaobserwować określone zjawiska konieczne będzie zbadanie długiego odcinka czasu. W takim przypadku wszystkie zjawiska, które istotne są lokalnie, tracą na znaczeniu, kiedy postawione są w szerszym kontekście.

## Technologia wykorzystana do stworzenia symulacji

### Wymagania dotyczące technologii

W rozdziale 2. oraz 3. przedstawiony został problem planowania produkcji i dystrybucji, a także umówiony został sposób, w jaki wymodelowana została symulacja komputerowa. Kolejnym krokiem jest postawienie określonych wymagań dotyczących technologii, jakie wykorzystane mogą być do stworzenia symulacji. Rozpocząć należy od przedstawienia listy funkcjonalności wymaganych od aplikacji:

1. **Interfejs** – Powinien umożliwiać przeglądanie list MRP w dowolnym momencie (w dowolnym kroku) symulacji. Interfejs użytkownika powinien również zapewniać sterowanie symulacją – tj. umożliwiać jej wstrzymanie, wznowienie, wykonanie pojedynczego kroku, czy n kroków. Ponadto z poziomu GUI[[22]](#footnote-23) istnieć powinna możliwość importowania danych z zewnętrznych (zasilających symulację) z plików CSV, a także eksportu wyników symulacji, by możliwe było obliczenie omawianych w rozdziale 4.1 kluczowych wskaźników efektywności.
2. **Baza danych** – Ponieważ zarówno ilość danych zasilająca symulację, jak i ilość danych generowanych w wyniku działania symulacji (takich jak zapis stanu magazynowego, czy ruchów towaru między ogniwami łańcucha dostaw w każdym kroku) jest znacząca, należy zadbać o właściwe ich przechowywanie. Niezasadne jest oczywiście trzymanie wszystkich danych wyłącznie w pamięci RAM komputera w czasie działania programu. Oczywiste wydaje się wykorzystanie do tego celu dedykowanej bazy danych. Z tego samego powodu wybrana technologia powinna umożliwiać szybkie i bezproblemowe połącznie z relacyjną bazą danych.
3. **Operacje IO** – Z uwagi na to, że niezbędne do wykonania symulacji jest zasilenie jej danymi firmy XYZ, a na jej wyjściu oczekiwane jest generowanie plików przeznaczonych do dalszej analizy, wybrana technologia wspierać musi łatwe operacje wejścia i wyjścia na plikach tekstowych (CSV, XML) i binarnych (XLSX). Rozumie się przez to istnienie dedykowanych bibliotek dostępnych na licencji open-source możliwych do wykorzystania przy imporcie, eksporcie, a także przy parsowaniu danych.
4. **Obiektowość** – Biorąc pod uwagę kompleksowość obiektów w symulacji (rozumianą przez mnogość i różnorodność ich atrybutów) znaczącym ułatwieniem jest mapowania świata rzeczywistego jako obiektów w języku programowania. Nie jest to jednak jedyny argument. Wykorzystanie obiektowego języka programowania umożliwia mapowanie obiektowo-relacyjne (ORM[[23]](#footnote-24)), które przyspiesza integrację aplikacji z serwerem bazy danych.
5. **Znajomość technologii** – Ostatnim, jednak nie najmniej ważnym, argumentem w dyskusji o technologiach możliwych do wykorzystania jest znajomość danej technologii (danego języka programowania oraz towarzyszących mu frameworków[[24]](#footnote-25)) przez autora niniejszej pracy.

### Przegląd i wybór technologii

Wymagania określone w rozdziale 3.2.1. nie dają jednoznacznej odpowiedzi na pytanie jaką technologię należy wykorzystać do stworzenia aplikacji. Dodatkowo niektóre z postawionych wymagań, takie jak obiektowość, czy łatwość operacji IO, są wspólne dla wielu wysokopoziomowych, współczesnych języków programowania. Biorąc jednak pod uwagę całokształt sprecyzowanych warunków wstępnych oraz ograniczając przestrzeń rozważań do najbardziej popularnych rozwiązań, wybór można ograniczyć do następujących technologii:

* Java + Swing + JDBC
* C# + WPF + .NET Sql Native Client
* C++ + Qt + ODBC

Biorąc pod uwagę aspekt czysto techniczny – wszystkie trzy rozwiązania umożliwią stworzenie aplikacji implementującej zarysowany w rozdziale 3.1. model. Różnią się one jednak od siebie znacząco, dlatego należy dokonać nieco głębszej ich analizy, by wybrać rozwiązanie najbardziej odpowiadające omawianemu problemowi.

Największą różnicą pomiędzy Javą a C++ jest sposób zarządzania pamięcią. W C++ to programista odpowiedzialny jest za alokowanie pamięci. Również w jego gestii pozostaje jej zwalnianie - poprzez instrukcję *delete*. Gdy zaalokowana pamięć pozostaje niezwolniona mimo, że nie jest już wykorzystywana, program (np. przy wywoływaniu określonej funkcji) pochłaniać będzie każdorazowo nowe zasoby, nigdy nie zwalniając tych nieużywanych. Doprowadza to wycieku pamięci - spadku wydajności, a w skrajnych wypadkach zawieszania się innych aplikacji, bądź też całego systemu operacyjnego. Problem ten nie jest obecny w najmniejszym nawet stopniu w Javie. Programista nie tylko nie musi martwić się o to, gdzie przechowywane są aktualnie przetwarzane dane, ale również nie ma obowiązku ich zwalniania. Rozwiązaniem tego problemu jest *Garbage Collector*. Identyfikuje on zasoby pozostające bez referencji i zwalnia je, umożliwiając alokację nowych danych.



Listing - Przykład wycieku pamięci w kodzie C++

Kolejną istotną różnicą jest sposób wykonywania samego kodu. C++ kompilowany jest do kodu binarnego, który uruchamiany jest bezpośrednio na procesorze. Java zaś kompilowana jest do kodu bajtowego, który uruchamiany jest następnie w JVM[[25]](#footnote-26). Każde z tych rozwiązań niesie ze sobą określone zyski i straty. Kod wykonywany bezpośrednio na procesorze z reguły przetwarzany będzie znacznie szybciej niż kod wykonywany na emulowanym procesorze Javy. Problem wydajności był dostrzegalny szczególnie w przeszłości, gdy szybkość procesorów i przepustowość pamięci była znacznie niższa niż obecnie. Obecnie nie stanowi to bardzo dużego problemu, jednak w przypadku gdy wymagane jest szybkie wykonywanie złożonych obliczeń i operowanie na dużych zbiorach danych, natywny kod binarny wykonywany będzie znacząco szybciej niż kod Javy.



Rysunek - Budowa JVM

Wykonywanie kodu na maszynie wirtualnej Javy niesie ze sobą jednak inne korzyści. Główną jest kompatybilność kodu bez względu na platformę. Raz napisana aplikacja może być uruchomiona na dowolnej platformie (sprzętowej, aplikacyjnej), jeżeli tylko istnieje odpowiednia dystrybucja JVM. Obecnie zaś trudno znaleźć urządzenie mobilne, czy system, które nie byłby wpierany przez Oracle[[26]](#footnote-27). Oznacza to łatwość portowania aplikacji, w myśl promowanego przez Sun Microsystems hasła *Write once, run anywhere*. Nie jest tak niestety w przypadku C++, który skompilowany musi być osobno na każdą platformę uruchomieniową.

Wieloplatformowość Javy oznacza również, że może być ona wykorzystana do tworzenia web serwisów oraz aplikacji webowych. Biorąc zaś pod uwagę obecne trendy rynkowe, nieuchronne wydaje się stopniowe przenoszenie wszystkich dotychczasowych rozwiązań do chmury. Możliwe jest wykorzystanie frameworków takich jak Wt[[27]](#footnote-28), aby użyć C++ do tworzenia aplikacji webowych, jednak w porównaniu z Javą lub środowiskiem .NET, wsparcie dla serwisów WWW jest tutaj znikome.

Ostatnim argumentem przemawiającym za wykorzystaniem Javy jest brak konieczności stosowania wskaźników, które bardzo często przysparzają programistom wielu problemów. Mogą to być chociażby omówione wcześniej wycieki pamięci.

Pod względem składni języka oraz sposobu zarządzaniem pamięcią C# okazuje się bardzo podobny do Javy. Programista Javy, bez żadnego problemu będzie w stanie zrozumieć kod napisany w C#, jak i odwrotnie. Nie oznacza to jednak, że są to języki bliźniacze, gdyż dzieli je wiele różnic. Pierwszym elementem obecnym w C#, a nieposiadającym żadnego odpowiednika w Javie jest LINQ[[28]](#footnote-29). W zdecydowany sposób upraszcza on operowanie na kolekcjach – wybieranie obiektów o określonych atrybutach, iterowanie po nich, modyfikowanie obiektów o wybranych cechach itd. Możliwość taka wielokrotnie oznacza oszczędność czasu programisty, a także upraszcza i sprawia bardziej czytelnym sam kod.



Listing - Przykład użycia LINQ w C#

Elementem wspólnym Javy i C# jest jednak to, że nie są one komplilowane do natywnego kodu przetwarzanego przez procesor. O ile jednak Java kompilowana jest do kodu bajtowego uruchamianego na maszynie wirtualnej, C# kompilowany jest do języka pośredniego (MSIL/CIL), który kompilowany jest w czasie rzeczywistym (poprzez kompilator JIT) do natywnych instrukcji procesora. Wadą tego rozwiązania w przeszłości była możliwość tworzenia aplikacji wyłącznie na platformę Windows. Obecnie jednak nie stanowi to żadnego problemu. Wraz z przedstawieniem .NET Core oraz Visual Studio Code możliwe jest wykorzystanie C# do tworzenia aplikacji zarówno na platformę Windows, jak i Linux oraz Mac. Zmiana ta wymusiła rzecz jasna otworzenie środowiska .NET Core, które obecnie dostępne jest jako open-source.



Rysunek - Kompilacja oraz wykonywanie kodu C#

Należy zwrócić uwagę również na same środowisko, w którym tworzona jest aplikacja. Visual Studio wraz z platformą .NET umożliwia tworzenie zarówno aplikacji webowych (Web API, MVC, SPA), jak i mobilnych (Xamarin, Ionic, Cordova) jak i desktopowych (WPF). Nie oznacza to może to aż takiej kompatybilności wskroś platform, jak w przypadku Javy, jednak ciągle pokrycie potencjalnych platform wymaga odnotowania.

Kolejną zaletą C# w stosunku do Javy jest obecność NuGet Package Managera. Ułatwia to w znaczącym stopniu zapytania do web serwisów, tworzenie aplikacji MVC, czy nawet wykorzystywanie technologii takich jak TTS[[29]](#footnote-30). W przypadku Javy w większości wypadków oznacza to konieczność wykorzystywania bibliotek i frameworków podmiotów trzecich. Do tego nieodzowne okazuje się używanie narzędzi do budowania/zarządzania referencjami takich jak Maven. W przypadku C# jest to zwyczajnie prostsze.

Ostatnią istotną różnicą pomiędzy trzema technologiami jest dziedziczenie klas oraz interfejsów. Zarówno w C++ jak i w C# możliwe jest wielokrotnie dziedziczenie klas, co znacząco ułatwia pracę, gdy programista pragnie wykorzystać wszystkie zalety obiektowego paradygmatu programowania. Zamiast tego Java oferuje wielokrotne dziedziczenie interfejsów, a także wewnętrze klasy anonimowe. Działają one w sposób bardzo podobny do domknięć w Scali. Od wersji Java 8 możliwe jest także wykorzystywanie wyrażeń lambda.

Podsumowując tę analizę należy dojść do wniosku, że równie prostymi i oferującymi potencjalnie najwięcej możliwości technologiami są równorzędnie Java oraz C#, pozostawiając C++ nieco w tyle jako język starszy i natywnie nie wspierający tak wielu rozszerzeń. Należy przy tym zaznaczyć, że ani Java ani C# nie posiada cech, które jednoznacznie wskazywałyby na wyższość jednego rozwiązania nad drugim w świetle naszkicowanych w rozdziale 3.1. wymagań. Do stworzenia symulacji została jednak wykorzystana Java z uwagi na to, że autor niniejszej pracy bardziej doświadczony jest w pracy z Javą i Swingiem, aniżeli w programowaniu w C# z wykorzystaniem Windows Presentation Foundation.

## Architektura aplikacji

Symulacja podzielona została na dwa zasadnicze elementy: aplikację oraz serwer bazodanowy. Dodatkowo pomiędzy aplikacją a serwerem MS SQL występuje dodatkowa warstwa w postaci interfejsu dostępu do danych.

Warstwa aplikacji odpowiedzialna jest za wszelkie manipulacje danych – począwszy od ich importowania, poprzez parsowanie, skończywszy na zasilaniu nimi bazy danych. Również tutaj przeprowadzana jest sama kalkulacja MRP, stąd też (za pośrednictwem mechanizmu do interpretacji scenariuszy testowych) sterowany jest przebieg samej symulacji. W samej aplikacji nie są jednak przechowywane żadne dane poza tymi, które są aktualnie przetwarzane. Wynik każdej przeprowadzonej symulacji (który stanowi zbiór elementów MRP) przechowywany jest w bazie danych.

Istotnym elementem jest wyodrębniony interfejs wymiany danych pomiędzy aplikacją a serwerem danych. Z uwagi na wygodę oraz czytelność kodu sama aplikacja operuje na generycznych obiektach takich jak *Shipment* czy *Delivery.* Przechowywane są one jako rekordy w bazie danych. By jednak możliwe było swobodne operowanie nimi na potrzeby kalkulacji/symulacji, a także z uwagi na potencjalnie wysoką powtarzalność kodu związanego z komunikacją z serwerem, wszelkie operacje bazodanowe zostały wydzielone. Interfejs danych (w oparciu o sterownik JDBC) odpowiedzialny jest za tłumaczenie danych na drodze z/do bazy, a także zbiorcze dokonywanie operacji bazodanowych. Jest to pośrednia realizacja mapowania obiektowo-relacyjnego, ale w uproszczonej formie, odpowiedniej dla tworzonej symulacji.



Rysunek - Architektura aplikacji

Ostatnią warstwą architektury jest serwer bazy danych. Wykorzystany został Microsoft SQL Server w wersji 2017. Częścią aplikacji są skrypty budujące niezbędne tabele oraz zasilające bazę danych danymi niezbędnymi do uruchomienia aplikacji. Każda z tabel posiada unikalny klucz główny lub kompozytowy klucz główny złożony z unikalnej kombinacji dwóch kolumn. Tabele w bazie danych podzielić można na trzy grupy:

1. Dane związane bezpośrednio z kalkulacją i przechowujące aktualny stan elementów MRP per ogniwo
2. *Master data*, dane stałe analizowanej sieci dystrybucyjnej
3. Dane statystyczne generowane po każdej kalkulacji MRP, służące do analizy przebiegu symulacji - obliczania KPI czy tworzenia graficznych reprezentacji procesów takich jak zmiana zapasu magazynowego w czasie

Diagram encji użytej bazy danych znajduje się w załączniku 7.4.

# Symulacja komputerowa

## Kluczowe wskaźniki efektywności

### Motywacja wyboru określonych wskaźników

TODO: Czemu każdy z nich jest istotny i jakie jest ich przełożenie na biznes

### Definicje

OFR (Order Fill Rate)

Gdzie:

– zamówienia zrealizowane

– wszystkie zamówienia

– zamówienia niezrealizowane

INV (Inventory Level)

Gdzie:

– zapas w ogniwie

DFC (Days Forward Coverage)

Gdzie:

– aktualny zapas produktu

– sprzedaż produktu przez ostatnie 30 dni

IST (Inventory Stock Turn)

Gdzie:

– sprzedaż produktu w czasie

– średni zapas produktu w czasie

DS (Dead Stock)

Gdzie:

– minimalny zapas produktu w czasie

NPI (Non-performing Inventory)

Gdzie:

– zapas towaru nierotującego

– średni zapas towaru w czasie

## Scenariusze testowe

### Zaburzenia występujące w rzeczywistych łańcuchach dostaw

### Scenariusz 1 - Błędna prognoza sprzedaży

* Metody tworzenia prognoz sprzedaży (SMO, PSC)
* Badanie wrażliwości łańcucha dostaw – jak zwiększenie się niedokładności prognozy sprzedaży wpływać będzie na wszystkie KPI w łańcuchu dostaw

### Scenariusz 2 – Zmiana progu replenishmentu, zmiana strategii replenishmentu

* Dokładne omówienie SS/ST, algorytmy wyznaczania SS, kontrolowanie safety
* Debata na temat tego, jakich parametrów bezpieczeństwa należy używać
* Badanie wrażliwości łańcucha dostaw – jak zmniejszenie/zwiększenie parametrów bezpieczeństwa wpływać będzie na wszystkie KPI w łańcuchu dostaw
* Czy zwiększenie poziomu serwisu o k procent warte jest zwiększenie średniego stanu o l procent?

### Scenariusz 3 – Zmiana minimalnego wolumenu produkcji

* Z czego wynikać mogą ograniczenia minimalnego wolumenu produkcji
* Debata na temat tego, czy zmiana częstości replenishmentów ma wpływ na średni zapas i poziom obsługi klientów

### Scenariusz 4 – Fałszywe dane w systemach planowania

* Źródła pochodzenia i potencjalne formy fałszywych danych (PDE)
* Badanie jak wiele czasu musi upłynąć od momentu wystąpienia zaburzenia do powrotu łańcucha dostaw do stanu wejściowego

## Dane testowe

### Informacje o danych testowych

### Pochodzenie danych testowych

# Wyniki

## Omówienie uzyskanych rezultatów

### Scenariusz 1

### Scenariusz 2

### Scenariusz 3

### Scenariusz 4

# Podsumowanie

## Weryfikacja postawionych tez

## Wnioski

# Załączniki

## Standard tworzenia sygnatur elementów MRP w symulacji

Z praktycznego punktu widzenia dobrą praktyką jest tworzenie sygnatur generowanych MRP elementów. Umożliwia to łatwe tworzenie referencji pomiędzy poszczególnymi elementami. Jest to istotne również z punktu widzenia użytkownika systemu ERP, który widząc sygnaturę o określonej budowie (zaczynającą się od odpowiedniego znaku alfanumerycznego, posiadającą ściśle określoną długość) jest w stanie zidentyfikować związany z nią MRP element. Odnosząc się również do architektury aplikacji, która oparta może być na relacyjnej bazie danych, jest to rozwiązania pożądane.

O ile implementacja sygnatur nie jest niezbędna w bieżącym zakresie pracy, może być bardzo pomocna w przypadku rozszerzenia ram symulacji.

Tabela - Spis sygnatur elementów MRP

|  |  |  |
| --- | --- | --- |
| **MRP Element** | **Pełna nazwa** | **Standard sygnatury** |
| Stock | Stock | Numeracja nie jest wymagana |
| ShipNt | Shipment Notification | 2xxxxxx : 7 cyfr, pierwsza to zawsze ‘2’, unikalna |
| PchOrd | Purchase Order | 4xxxxxx : 7 cyfr, pierwsza to zawsze ‘4’, unikalna |
| PrcOrd | Process Order | 6xxxxxx : 7 cyfr, pierwsza to zawsze ‘6’, unikalna |
| PlOrd | Planned Order | 8xxxxxx : 7 cyfr, pierwsza to zawsze ‘8’, unikalna |
| QMLot | Quality Inspection Lot | 0xxxxxx : 7 cyfr, pierwsza to zawsze ‘0’, unikalna |
| Safety Stock | Safety Stock | Numeracja nie jest wymagana |
| Order | Order | 1xxxxxx : 7 cyfr, pierwsza to zawsze ‘1’, unikalna |
| Deliv | Delivery | 3xxxxxx : 7 cyfr, pierwsza to zawsze ‘3’, unikalna |
| DepReq | Dependent Requirement | 5xxxxxx : 7 cyfr, pierwsza to zawsze ‘5’, unikalna |
| PlORel | Planned Order Release | 7xxxxxx : 7 cyfr, pierwsza to zawsze ‘7’, unikalna |
| IndReq | Independent Requirement | 9xxxxxx : 7 cyfr, pierwsza to zawsze ‘9’, unikalna |

## Standard tworzenia znaczników czasu elementów MRP w symulacji

Tabela - Spis znaczników czasu elementów MRP

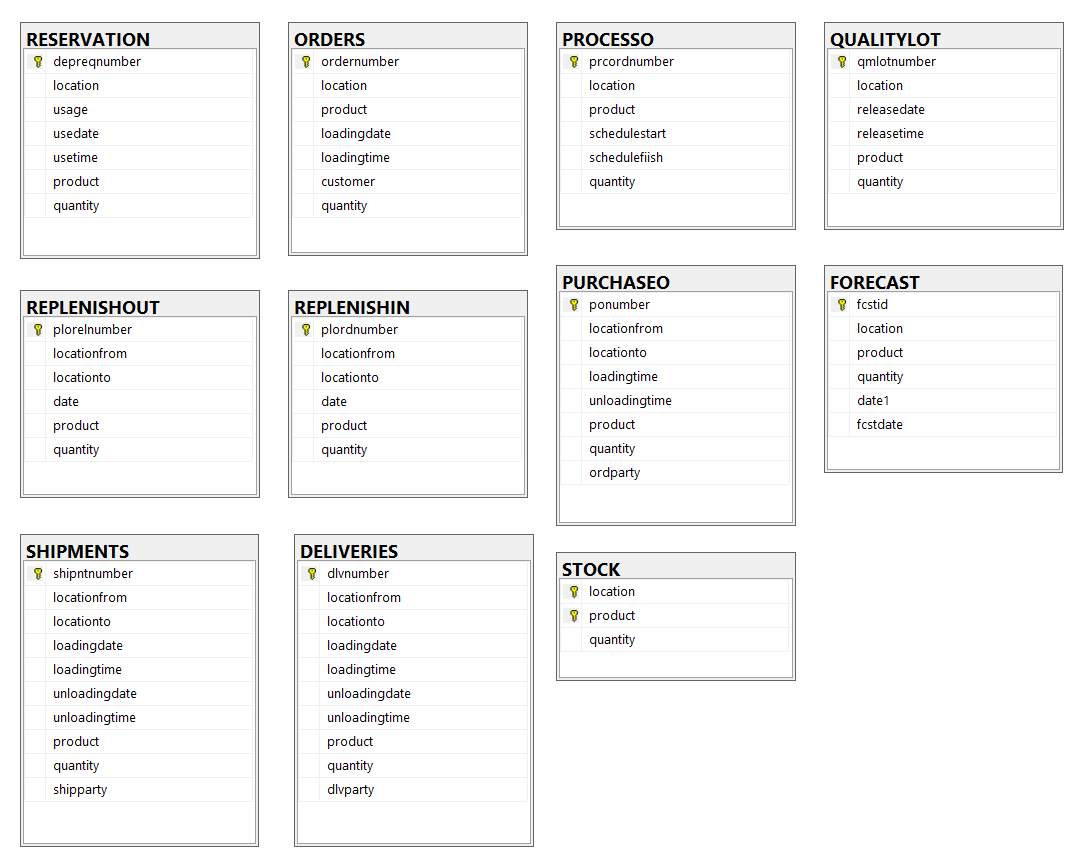
|  |  |  |
| --- | --- | --- |
| **MRP Element** | **Pełna nazwa** | **Standard znacznika czasu** |
| Stock | Stock | TODO |
| ShipNt | Shipment Notification | TODO |
| PchOrd | Purchase Order | TODO |
| PrcOrd | Process Order | TODO |
| PlOrd | Planned Order | TODO |
| QMLot | Quality Inspection Lot | TODO |
| Safety Stock | Safety Stock | TODO |
| Order | Order | TODO |
| Deliv | Delivery | TODO |
| DepReq | Dependent Requirement | TODO |
| PlORel | Planned Order Release | TODO |
| IndReq | Independent Requirement | TODO |

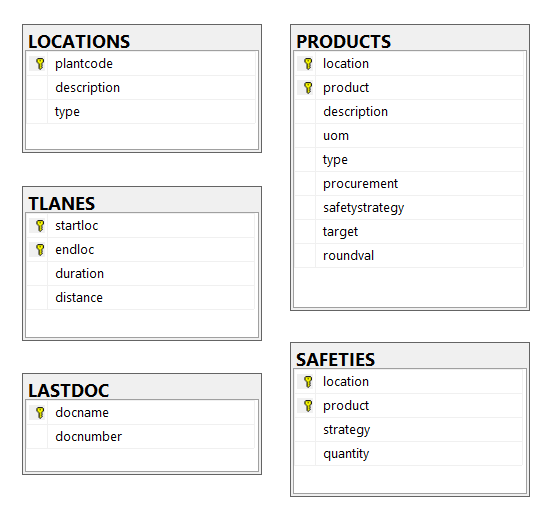
## Standard tworzenia plików XML ze scenariuszami testowymi

TODO: Opracować standard plików XML ze scenariuszami testowymi



## Tabele w bazie danych





# Bibliografia

Boon Ping Gan, Li Liu, Sanjay Jain, Stephen J Turner, Wentong Cai i Wen-Jing Hsu. *Distributed Supply Chain Simulation Across Enterpirse Boundaries.* 2000.

Jayshankar M Swaminathan, Steven F Smith i Norman M Sadeh. „Modeling Supply Chain Dynamic: A Multiagent Approach.” *Decision Sciences*, 1998: 607-633.

Matthias Holweg i John Bicheno. „Supply chain simulation } a tool for education, enhancement and endeavour.” *International Journal Of Production Economics*, 2002: 163-175.

Rodolfo Garcıa-Flores i Xue Zhong Wang. „A multi-agent system for chemical supply chain simulation and management support.” *OR Spectrum*, 2002: 343–370.

TODO: Uzupełnić bibliografię o przeczytane I przejrzane pozycje

1. KPI – Key Performance Indicators, wskaźniki wykorzystywane do pomiaru stopnia realizacji celów biznesowych [↑](#footnote-ref-1)
2. FMCG – Fast Moving Consumer Goods, zróżnicowane dobra sprzedawane często, o względnie niewielkiej wartości i wysokich wolumenach sprzedaży, tj. artykuły higieniczne czy środki czystości [↑](#footnote-ref-2)
3. MAS – Multi Agent System, system złożony z komunikujących i współpracujących między sobą agentów, realizujących określone cele, częstokroć wykorzystywany w do modelowania rozproszonych i złożonych obliczeniowo problemów [↑](#footnote-ref-3)
4. KQML - Knowledge Query and Manipulation Language, język i protokół komunikacji pomiędzy agentami i systemami opartymi o dane, umożliwiający wzajemną komunikację [↑](#footnote-ref-4)
5. SCM – Supply Chain Management, kontrola przepływu materiałów i ładunków między ogniwami łańcucha dostaw, najczęściej realizowana przy pomocy dedykowanego oprogramowania [↑](#footnote-ref-5)
6. MRP – Material Requrements Planning – ogół procesów planowania zapotrzebowania materiałowego, sterowania produkcją i zapasami [↑](#footnote-ref-6)
7. SAP – zintegrowane, modułowe oprogramowanie służące do zarządzania zasobami przedsiębiorstwa, stworzone przez SAP SE [↑](#footnote-ref-7)
8. Replenishment – uzupełnienie zapasów, dostawa [↑](#footnote-ref-8)
9. RCCP – Rough Cut Capacity Planning – weryfikacja dostępnych zasobów względem planu idealnego, podstawowy element planowania długoterminowego [↑](#footnote-ref-9)
10. BOM – Bill Of Materials, struktura produktu określająca zestawienie półproduktów, komponentów i surowców wraz z ilościami, które niezbędne są do wytworzenia jednostki produktu końcowego [↑](#footnote-ref-10)
11. ERP – Enterprise Resource Planning, zbiór systemów informatycznych wspomagających całościowe zarządzanie przedsiębiorstwem oraz jego zasobami [↑](#footnote-ref-12)
12. APS – Advanced Planning and Scheduling – rozszerzenie standardowych procesów zintegrowanego planowania produkcji oparte o określone rozwiązanie IT, np. OMP Plus [↑](#footnote-ref-13)
13. APO – Advanced Planner and Optimizer, rozszerzenie SAP R/3 wykorzystywane do planowania dystrybucji w rozproszonym łańcuchu dostaw [↑](#footnote-ref-14)
14. PIPO – Phase In Phase Out, para produktów, które rozróżnialne są z punktu widzenia planowania (np. posiadają różne BOM, różne metody paletyzacji), lecz nierozróżnialne są dla klienta (posiadają taki sam kod EAN) [↑](#footnote-ref-15)
15. GI-GO – garbage in – garbage out, niskiej jakości dane wejściowe generować będą niskiej jakości dane wyjściowe, nawet jeżeli przetwarzane są przez najlepiej zaprojektowane i dokładne algorytmy [↑](#footnote-ref-16)
16. customer facing DC – centrum dystrybucyjne, za pośrednictwem którego realizowane są zamówienia klienta [↑](#footnote-ref-17)
17. VAS – value added services, manipulacje towarem, usługi które służą dodaniu lub zmianie wartości towaru jak np. umieszczenie naklejek, dodanie darmowej próbki innego produktu, stworzenie paczki wielu produktów oferowanej jako jedno dobro klientowi itd. [↑](#footnote-ref-18)
18. EAN – ang. European Article Number, numeryczny modularny kod towaru z zapisaną sumą kontrolną służący do identyfikacji towaru, wykorzystywany najczęściej w logistyce magazynowej i sprzedaży detalicznej [↑](#footnote-ref-19)
19. TMS – Transport Management System, elektroniczny system zarządzania transportem, którego główną kompetencją jest śledzenie statusu oraz kontrolowanie pojazdów znajdujących się w tranzycie [↑](#footnote-ref-20)
20. YMS – Yard Management System, elektroniczny system zarządzania pojazdami na terenie zakładu przemysłowego, główną kompetencją jest koordynacja dostępu do doków, parkingów i wag [↑](#footnote-ref-21)
21. MAPE – Mean Absolute Percentage Error, średni błąd odchylenia standardowego [↑](#footnote-ref-22)
22. GUI – Graphical User Interface, graficzny interfejs użytkownika, powłoka graficzna pozwalająca sterować aplikacją [↑](#footnote-ref-23)
23. ORM – Object-Relational Mapping, sposób odwzorowania obiektowej architektury systemu informatycznego na bazę danych (lub inny element systemu) o relacyjnym charakterze (Mapowanie obiektowo-relacyjne 2017) [↑](#footnote-ref-24)
24. framework – ramy służące do budowy aplikacji, dostarczają szkielet aplikacji oraz definiują mechanizm jej funkcjonowania, zawierają zestaw generycznych komponentów oraz bibliotek ułatwiających implementację określonych działań [↑](#footnote-ref-25)
25. JVM – Java Virtual Machine, maszyna wirtualna Javy wyposażona w środowisko uruchomieniowe kodu bajtowego Javy, rozróżniane są dwie główne dystrybucje: JRE (Java Runtime Environment) będąca jedynie środowiskiem uruchomieniowym oraz JDK (Java Development Kit) zawierająca narzędzia dla programistów [↑](#footnote-ref-26)
26. Oracle – amerykańskie przedsiębiorstwo, które przejęło Sun Microsystems odpowiedzialne za stworzenie języka Java [↑](#footnote-ref-27)
27. Wt, Witty, framework C++ umożliwiający tworzenie web serwisów, posiada zintegrowany serwer aplikacji, http://www.webtoolkit.eu/ [↑](#footnote-ref-28)
28. LINQ – Language Integrated Query – technologia .NET umożliwiająca zadawanie zapytań na obiektach, przeglądanie kolekcji w prosty sposób przypominający język zapytań SQL [↑](#footnote-ref-29)
29. TTS – ang. Text To Speech, synteza mowy, przetwarzanie teksu na mowę [↑](#footnote-ref-30)