

[サプライチェーンの原理と経営]

モデル解説編

© 森田道也, 2004.6.25

- 1 . 第 2 章のモデル解説
 - 例 2 - 1 店舗モデル 1
 - 例 2 - 2 店舗モデル 2
 - 例 2 - 3 目標追求モデル
 - 例 2 - 4 キャッシュフロー・モデル
- 2 . 第 3 章のモデル解説
 - 例 3 - 2 ロット生産モデル 1
 - 例 3 - 5 ロット生産モデル 2
- 3 . 第 4 章のモデル解説
 - 例 4 - 10 経済的発注公式モデル
- 4 . 第 5 章のモデル解説
 - 例 5 - 3 小売店在庫維持モデル 1
 - 例 5 - 4 小売店在庫維持モデル 2
- 5 . 第 6 章のモデル解説
 - 例 6 - 2 プッシュ方式モデル
 - 例 6 - 3 プル方式モデル
 - 例 6 - 4 独立方式モデル
 - 例 6 - 5 プッシュ方式の適正確保量への不適應モデル
 - 例 6 - 6 プル方式の適正確保量への不適應モデル
 - 例 6 - 7 リードタイム・モデル
 - 例 6 - 8 ブルウィップ・モデル
 - 例 6 - 9 ブルウィップ・モデル 2
- 6 . 第 7 章のモデル解説
 - 例 7 - 1 ボトルネック・モデル 1
 - 例 7 - 4 ガソリンステーション・モデル
 - 例 7 - 5 ボトルネック・モデル 2
 - 例 7 - 6 ボトルネック・モデル 3 (プッシュ方式)
 - 例 7 - 7 ボトルネック・モデル 4 (独立方式)
 - 例 7 - 8 ボトルネック・モデル 5 (プル方式)

第2章のモデル解説

35 頁 例 2 - 1 店舗モデル 1

モデルの意図

本モデルは靴下を仕入れ、販売している店舗の操業をシミュレーションしている。

モデルの構造

シミュレーション・モデルの構造は下図のようになっている。

操業条件は以下の通りである。

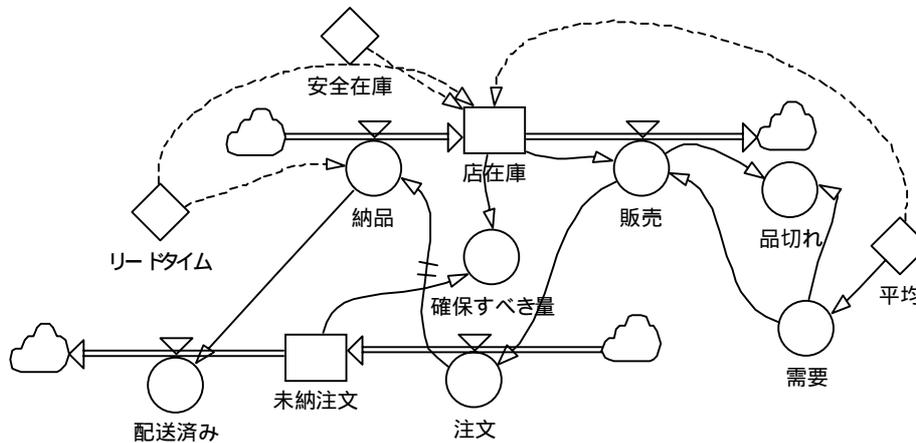
毎日一定の「**平均**」(20 足)の「**需要**」がある。

店舗が注文して卸業者から配送されるまで7日間の「**リードタイム**」がある。7日目の夕刻に「**納品**」される。

店舗は毎日終業した時点で、その日に「**販売**」量(売れた分)を「**注文**」する。すなわち、発注間隔は1日になる。

「**確保すべき量**」は「**店在庫**」と「**未納注文**」の和になる。

「**品切れ**」は「**需要**」が「**販売**」を超えた分である。



モデルを構成する要素とその関わりを解説しておく。

まず四角のレベル(ないしストック)変数については、シミュレーションが開始される時の状態を設定しておく必要がある。設定した状態は初期条件あるいは初期値という。

モデルには2つのレベル変数がある。「**店在庫**」と「**未納注文**」である。それらの初期値を確認しておく。まず、「**店在庫**」の上にカーソルを置いて、ダブルクリックする。以下のような「**店在庫**」Level Symbol Properties画面になる。その**Definition**欄には下記のようなステートメントがある。

平均*(リードタイム+1)+安全在庫

これは、「平均」と「リードタイム」に1を加えたものを乗じ、それに「安全在庫」を加えた値という意味である。「平均」は菱形の形状で表わされた要素（パラメータあるいはある定数で定義される要素を表現する）の値で、「平均」をダブルクリックして「平均」Auxiliary Symbol Properties 画面にすると Definition 欄に 20 となっていることがわかる。

すなわち、「店在庫」の初期値を計算する場合に、「平均」には 20 という値が使われる。「リードタイム」に1を加えるのは、注文リードタイムの「リードタイム」と注文間隔1日（1日に1回の注文だから）を加えて総リードタイムとするためである。「リードタイム」はやはり定数として定義されていて菱形で表現される。その値は「平均」と同様の手順で見ることができる。その値は7になっている。最後の「安全在庫」も定数で与えられている。その値は現在、0となっている。需要が変動しないためである。

最終的に「店在庫」の初期値は $20 \times (7+1) + 0 = 160$ である。

あと1つのレベル変数「未納注文」の初期値はダブルクリックして確認すると0である。シミュレーションがスタートする段階ではまだ注文残がないという状況を想定している。

レベル変数を変化させる要素がレイト変数（矢印と丸が一緒になった形状で表現する）である。モデルでは4つある。「販売」、「納品」、「注文」、「配送済み」である。それらはある1日の量である。「店在庫」はその日に売れて出ていく「販売」と注文して納品されてくるその日の「納品」で変化する。前者の方が多い場合には「店在庫」は減少し、逆に少ない場合には「店在庫」は増加する。「未納注文」は「注文」と「配送済み」によって変化する。「未納注文」は「店在庫」の変化と同様に、「注文」が「配送済み」より多い場合に増え、逆に「注文」が「配送済み」より少ない場合には減少する。

これら4つのレイト変数はどのようにして定義されているかを見る。それらをダブルクリックすると以下のように定義されていることがわかる。

「販売」: $\text{MIN}(\text{店在庫}, \text{需要})$ （Window 2000 以前の場合には $\text{MIN}(\text{店在庫}; \text{需要})$ ）

「納品」: $\text{DELAYPPL}(\text{注文}, \text{'リードタイム'}, 0)$

（Window 2000 以前の場合には $\text{DELAYPPL}(\text{店在庫}; \text{リードタイム}; 0)$ ）

「注文」: 販売

「配送済み」: 納品

まず、 $\text{MIN}(\text{店在庫}, \text{需要})$ の意味であるが、 $\text{MIN}(a, b)$ となっている場合にはaとbの小さい方が「販売」になるということである。すなわち、「需要」がいくらあっても、現在ある「店在庫」よりも多く売ることはできないからである。逆に「需要」が「店在庫」よりも少なければ「需要」が「店在庫」になる。 $\text{MIN}(a, b)$ は組み込み関数として装備されている。

次に、 $\text{DELAYPPL}(\text{注文}, \text{'リードタイム'}, 0)$ であるが、この意味は、「注文」量の値が「リードタイム」だけ遅れて「納品」量になるという意味である。括弧内の最後の0はシミュレーションが始まる初期段階において過去の値がいくつであったかを規定するもので、0という場合には注文していなかったことを意味する。遅れを持つ場合には、このような過去の値の定義をするわけである。0のところは何もいれなかったならば、モデルではデフォルト扱いになり、注文の初期値が過去の値として自動的な計算がされる。

$DELAYPPL(a, b, c)$ は組み込み関数として装備されているが、 a が適用される量、 b が遅れの長さ、 c は過去の量を意味する。

「注文」と「配送済み」はそのまま「販売」と「納品」の値として定義されるということである。

補助変数(丸形状だけで表現する)はモデルでは3つある。「品切れ」、「確保すべき量」、「需要」である。それぞれの定義は以下のようにになっている(見方は前述の通り、それぞれの上にカーソルを置いてダブルクリックして **Definition** 欄を見る)。

「品切れ」: $MAX(0, 需要 - 販売)$

「確保すべき量」: 未納注文 + 店在庫

「需要」: 平均

「品切れ」の定義で用いられている $MAX(0, 需要 - 販売)$ は「需要」から「販売」を引いたものと、0の大きい方が「品切れ」となることを意味する。先の $MIN(a, b)$ と逆の意味の組み込み関数である。「確保すべき量」は「未納注文」と「店在庫」の和として定義されている。「需要」は「平均」そのままになる。

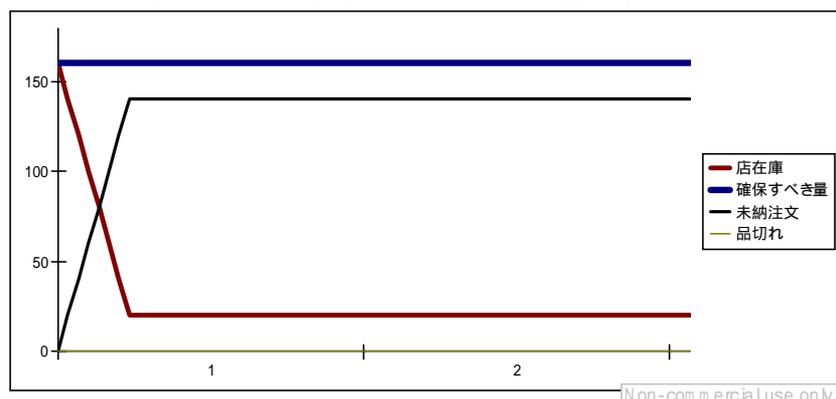
定数は菱形の形状の記号で表現する。モデルには「平均」、「安全在庫」、「リードタイム Δ 」が定数として用いられている。

操作の基本

モデルにおける数字などを変化させる操作について説明しておく。例えば、「リードタイム Δ 」の値を変更する場合には、上の Powersim モデル画面で、リードタイムと命名された菱形の印の上にカーソルを置いてダブルクリックし、「リードタイム Δ Auxiliary Symbol of Properties」画面を開く。その画面にある **Definition** の欄に半角英数字で入れたい数字を打ち込む。値を入れた後で、下の **OK** を1回クリックする。その値でシミュレーションするときには、上の **Simulation** を1回クリックし、ドラッグして開いたものから **Play** をクリックする。

縦軸のスケール上限を変化させる場合には、上のグラフの右側にある線の説明の上あたりをダブルクリックし、**Time Graph Control of Properties** の画面にし、その上にある **Value Axis** を1回クリックする。出てきた画面の左下の **Scale** で下限値(**Minimum**)と上限値(**Maximum**)に希望する値を半角英数字で入力する。なお、自動設定にするには、それらの左側にある **Auto** を選択する。下にある **Auto** を1回クリックする。

結果はフロー図の下にある次のようなグラフとして出る。



シミュレーション演習のポイント

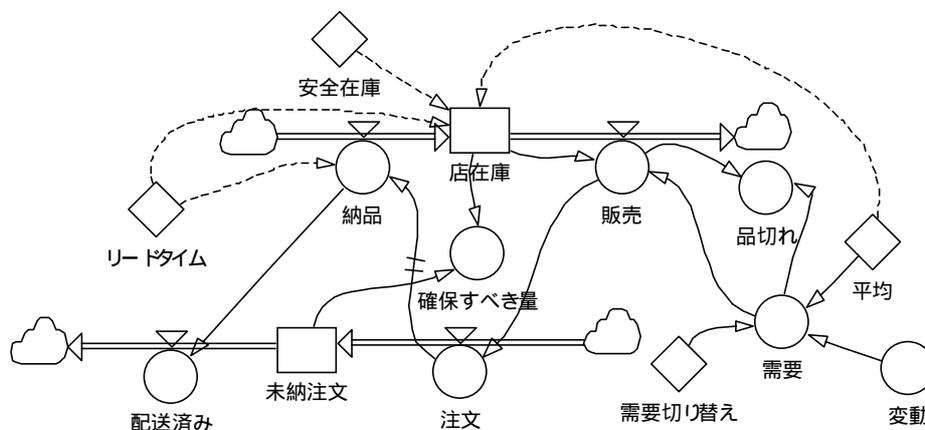
リードタイムを現在の7日から長く、あるいは短くして店舗在庫、未納注文（オン・オーダー量）確保すべき量などの動きを見ることでリードタイムと投入資源量の間係を理解する。

モデルの意図

本モデルは本質的には店舗モデル 1 と同じである。違いは、需要のパターンにある。すなわち、「毎日一定の 20 足の需要がある」のではなく、上下に周期的に変化するパターンになっている。

モデルの構造

シミュレーション・モデルの構造は下の図のようになっている。



店舗モデル 1 との違いは、「**変動**」と「**需要切り替え**」という要素が入って、それらが「**需要**」を規定していることである。丸い印の「**需要**」にカーソルを置いてダブルクリックすると「**需要**」の「**需要**」 Auxiliary Symbol of Properties 画面となるが、その **Definition** 欄を見ると需要が $\text{平均} + \text{変動} * (\text{需要切り替え})$ と定義されていることがわかる。

「**平均**」は店舗モデル 1 の 20 足である。また「**変動**」は需要が変動する大きさの値である。その値としてプラスとマイナスの数字が入っている。すなわち、「**変動**」内にカーソルを置き、ダブルクリックして、「**変動**」 Auxiliary Symbol of Properties 画面を開くと、以下のような定義になっていることがわかる。

GRAPH(TIME,0,1,{0,1,2,3,4,5,4,3,2,1,0,-1,-2,-3,-4,-5,-4,-3,-2,-1,0,1,2,3,4,5,4,3,2,1,0,-1,-2,-3,-4,-5,-4,-3,-2,-1,0,1,2,3,4,5,4,3,2,1,0,-1,-2,-3,-4,-5,-4,-3,-2,-1,0,1/Min:0;Max:1/})

{ } 内で//までの数字が変動部分になる。最初(0 時点、すなわち現時点)は 0 で、毎日 1 足ずつ増え、5 足まで増えると今度はそれをピークに 1 足ずつ減少する。減少は最高 5 足までである。

「**需要切り替え**」は 0 または 1 の値によって上の「**変動**」分を平均に加えるかどうかをコントロールする。もしそれを 0.5 と設定すれば、変動分が半分の大きさになる。

シミュレーション演習のポイント

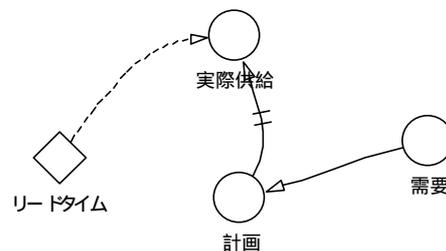
例 2 - 1 と同じ。

モデルの意図

市場の需要の動きに応じて供給する行動を考える。目指す需要の動きが判明してそれを満たすように供給するが、リードタイムの遅れによってなかなか供給したい需要量を達成できないことを理解する。

モデルの構造

モデルは下のような単純な構造になっている。



操業条件は、

供給「**リードタイム**」が30日。

「**需要**」は平均が1日当たり100個、振幅がピークでプラス、マイナス30個（需要個数とすれば最高が130個、最少が70個になる）で周期的に変化するサイン曲線である。その周期は60日である。言い換えれば、100個から130個まで増加し、それから100個を通過して70個まで減少し、それが100個に再度戻ってくるまでの期間が60日である。

「**計画**」は「**需要**」を実現するように「**需要**」と等しく置かれている。

需要パターンについて解説を加えておく。「**需要**」をダブルクリックして「**需要** Auxiliary Symbol of Properties」画面とすると、その**Definition**欄に下のような式が定義されている。

$SINWAVE(30, 60)+100$

$SINWAVE(30,60)$ はサイン曲線を定義する関数で、**30**が上下へのそれぞれ振幅を意味し、**60**は日数で、ピークからピークまでの周期を定義する。

シミュレーション演習のポイント

リードタイムが長くなると目標の需要の動きをなかなか捉えられないことを理解する。ただし、周期が60日で上下に等しく変化することを利用して60というリードタイムにして波形が一致するとしてもこの例では意味がない。

モデルの意図

小売店舗の「**マージン**」(「**売上高**」から「**仕入支払高**」を差し引いた粗利益)が注文「**リードタイム**」によって左右されることを理解する。注文「**リードタイム**」は仕入(支払い)と販売(収入)の時間的ずれを生み出すことを理解する。

モデルの構造

モデルは以下のような構造になっている。操業条件は、当初は毎日 100 個の「**需要**」を仮定し、次にその「**需要**」が「**平均**」100 個、それを中心に最高 130 個、最低 70 個まで周期的に変化するサイン曲線になる需要を想定する。周期は 90 日としている。

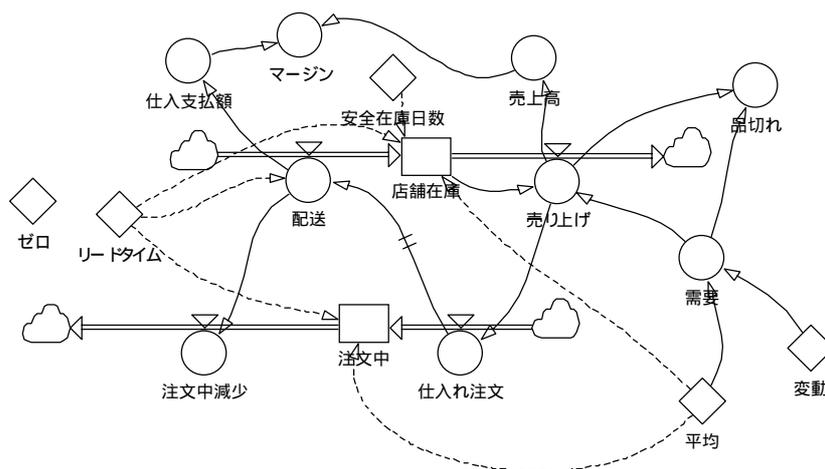
その商品の販売単価を 1 とし、仕入単価は 0.7 とする。

売れた個数(「**売り上げ**」)を営業終了後の夕刻に注文し、「**仕入れ注文**」とする。従って発注間隔は、1 日になる。

その商品が「**配送**」されると、配送時点でその仕入額を支払う。また「**売上高**」は売り上げた時点に入金する。

「**需要**」がプラス、マイナス 30 個で変動するので、それに備える在庫を「**安全在庫日数**」を定義して保有できる。ここでは 3 日分を想定する。

「**マージン**」は「**売上高**」から「**仕入支払額**」を引いて求める。



「**変動**」が 0 とされると毎日 100 個の一定需要になる。0.3 とすると最高 130 個、最少 70 個まで振れるサイン曲線需要になる。また店舗在庫が初期段階で 100 個しかないとサイン曲線で需要が変化する場合に品切れが出るので 3 日分の需要に当たる分を安全在庫(平均以上になる需要の備える在庫)として抱えているようにしている。従って、サイン曲線ではない一定の需要を想定するとき(「**変動**」を 0 にする)には、「**安全在庫日数**」の

Definition 欄にある 0 と置くことになる。「ゼロ」は図で示すときに、マージンが 0 になる水準を参考としてわかるように入れている。

シミュレーション演習のポイント

今までと同じように、リードタイムがマージンの動きに与える影響を理解する。リードタイムがキャッシュ・フローを変化させることを確認し、リードタイムの経営的意味合いを理解する。

第3章のモデル解説

69頁 例3-2 ロット生産モデル1

モデルの意図

原材料から完成品までの生産の流れを考え、各生産段階（活動）で終了したものを後の生産段階へと一度に送り出す量（ロット）が全体の生産終了時間（リードタイム）にいかに関与するかを理解する。

モデルの構造

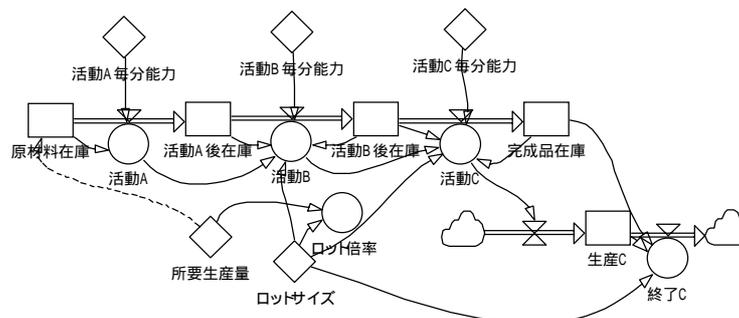
モデルの操作条件は下の通りである。

生産は原材料在庫から原材料を取り出して、まず「活動A」を行う。次にAが終了したもの、すなわち「活動A後在庫」にあるものを取り出して「活動B」が加工する。終わったものは「活動B後在庫」に入り、最後の「活動C」がそこにあるものを取り出して加工する。Cが終了したものは「完成品在庫」に入る。

活動A、B、Cの加工能力は1分当たりの加工量で定義されている。活動Aの能力は「活動A毎分能力」で表わされ、10/60個、活動Bが12/60個、活動Cが8/60個である。すなわち、Aは1時間に10個、Bは1時間に12個、Cは1時間に8個加工できる。

加工すべき総量（「所要生産量」）は1,200個とし、あらかじめ「原材料在庫」にその分はあるとする。

各活動ではその前にある在庫（例えば活動Bであれば活動A後在庫）に「ロットサイズ」以上あれば加工を開始できる。それ未満であれば開始できない。ただし、「ロットサイズ」の量が「所要生産量」を割り切れない（端数が出る）場合には、最後のその端数個数分については「ロットサイズ」よりも少なくとも加工開始はできる。例えば、「所要生産量」が1,200個、「ロットサイズ」が500個とすると、1,000個までは「ロットサイズ」通りに行くが、第3ロットになると200個しか在庫にはない。その200個は200個になれば後の活動が開始できることになる。



各活動の毎分能力を変化させたい場合には、例えば「活動 A 毎分能力」にカーソルを置いてダブルクリックして、「活動 A 毎分能力」Auxiliary Symbol of Properties 画面にし、Definition 欄に設定したい能力を半角英文字で数字を入力する。「ロットサイズ」の大きさを変化させる場合も同様の手順を踏む。

シミュレーション演習のポイント

「ロットサイズ」の大きさを 1,200 個以下で変化させて 1,200 個全部が完成品に入るまでの時間の変化を理解する。

モデルの意図

2つの、平行して走る生産工程が最後に合流して組み立てラインに入るという構造で、ロットサイズによって生産終了時間がいかに決まるかを理解する。この場合には2つの工程のどちらか長い方がその時間を規定することに注意してもらうことが重要である。すなわちクリティカル・パス概念を理解することが目的である。

モデルの構造

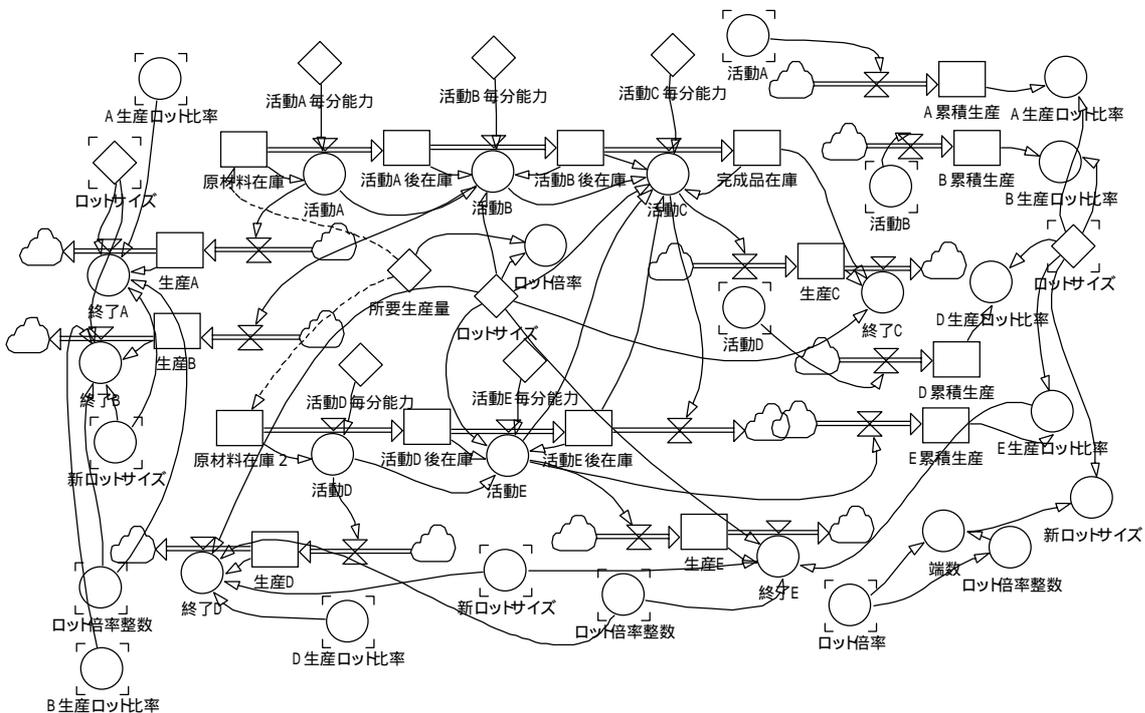
モデルの構造は以下のようにになっている。

前のロット生産モデル1との違いは以下のようにになる。

「活動A」と「活動B」に平行する「活動D」と「活動E」の工程が付加される。

新たな「活動D」の毎分加工数は8/60個、「活動E」のそれは10/60個である。上の工程にある活動A、B、Cの能力はそれぞれ10/60個、12/60個、8/60個でロット生産モデル1と変わらない

最後の「活動C」で2つの平行する工程から来たものが一緒に合わさって加工され、完成品ができ、「完成品在庫」として積みあがる。



ロット生産モデル1と異なる点は、活動DとEからなる別の工程が付加されたこと、そしてそれぞれの活動の生産量が変わるようにそれら生産量の計算を行う部分が拡充されたことだけである。それには案外複雑な処理が必要であるが、モデルの意味合いを理解するにはこれら技術的な処理方法に関する問題は関係がないので詳しい説明は省略する。

シミュレーション演習のポイント

ロットサイズの変更による生産終了時間の変化を理解することが基本的なポイントになるが、複数の平行的工程におけるクリティカル・パスがどこになるかを見ることも焦点である。各活動の生産能力を変化させてそのパスがどう変化するかも見てほしい。

第4章のモデル解説

122頁 例4-10 経済的発注公式モデル

モデルの意図

経済的発注公式は在庫理論では基礎的である。発注量に応じて相反する動きをする主要な二つの費用、すなわち、在庫維持（保有）費用と発注費用の性質を理解し、それらの和を最小にするような発注量を決めるという論理を理解してもらうことが目的である。

モデルの構造

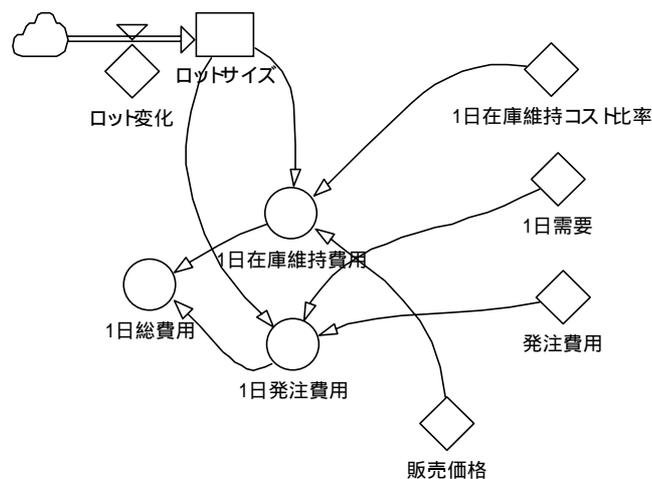
モデルはロットサイズの大きさを1個から5個ずつ増やしていくにつれて在庫維持費用と発注費用、およびそれらの和がどうなるかをシミュレーションする構造になっている。増やす刻み（現在は5）は「**ロット変化**」によって規定する。なお、結果の図の縦軸スケールは2つの費用の動きが良くわかるように、上限を低くしている。

作業条件は以下の通りである。

「**発注費用**」は1回当たり**30,000**円。「**1日在庫維持コスト比率**」は製品単価の**5%**だけかかると想定している。

「**1日需要**」（1日当たりの需要量）は一定の**10**個である。

「**1日総費用**」は「**1日在庫維持費用**」と「**1日発注費用**」の和である。



シミュレーション演習のポイント

「**1日在庫維持コスト比率**」、「**発注費用**」を変化させて最小費用のロットサイズがどう動くかを理解する。また両方の費用の和が最小になる点が谷の底のような位置に来ることを理解すると同時に、底が1つだけということにも注意する。

第5章のモデル解説

178頁 例5-3 小売店在庫維持モデル1

モデルの意図

小売店舗の仕入れ問題を例に、リードタイムがあると、小売りに在庫に関して理想在庫堅持政策による店作りが難しいことを理解する。

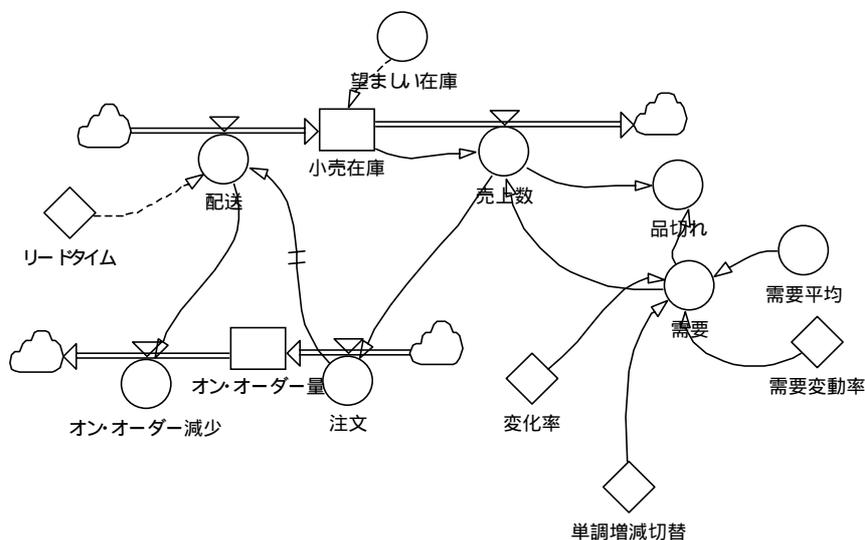
モデルの構造

例2-1 店舗モデル1 と殆ど変わらない。新しい点は「**望ましい在庫**」が付加されたことと、需要パターンにバリエーションが新しくつけられるようにしていることである。「**オン・オーダー量**」とは未納注文量のことであり、「**オン・オーダー減少**」は配送済みの量である。店の操業条件は、以下の通りである。

注文「**リードタイム**」は例2-1 店舗モデル1 と同じ**7日**である。終業後に売れた数量を卸に注文する。従って、発注間隔は**1日**になる。

需要は1日当たり平均**20着**で一定というのが標準で、それに最高が**25着**、最低が**15着**という周期的に変化するパターンと単調に増減するパターンの4種類がある。なお、周期パターンでは**180日**と長くなっている。

店では営業開始前に「**望ましい在庫**」として必ず**40着**店に置いておくことを理想としている。お客が選択できるようにするためである。



「望ましい在庫」は「小売在庫」の初期値として入れられている。需要パターンは4種類で、一定の20着のパターン、最高25着から最低15着まで周期的に増減するパターン、単調に一年で需要がなくなるパターン（「変化率」が $-20/365$ の値）、半年で50%増加するパターン（「変化率」が $10/365$ の値）である。各パターンを入れる場合は以下のような操作を行う。

一定需要20着のパターン：「単調増減切替」と「需要変動率」を両方とも0とする。

周期パターン：「単調増減切替」を0とし、「需要変動率」を0.25とする。

一定減少パターン：「単調増減切替」を1と設定し、「変化率」を $-20/365$ とする。

一定増加パターン：「単調増減切替」を1と設定し、「変化率」を $10/365$ とする。

シミュレーション演習のポイント

まずは需要パターンを上のように変化させて小売在庫が理想の40着を維持できるかを見る。次に、リードタイムを変化させてそれぞれの需要パターンに応じて理想在庫堅持政策に対していかなる影響が加わるかを理解する。

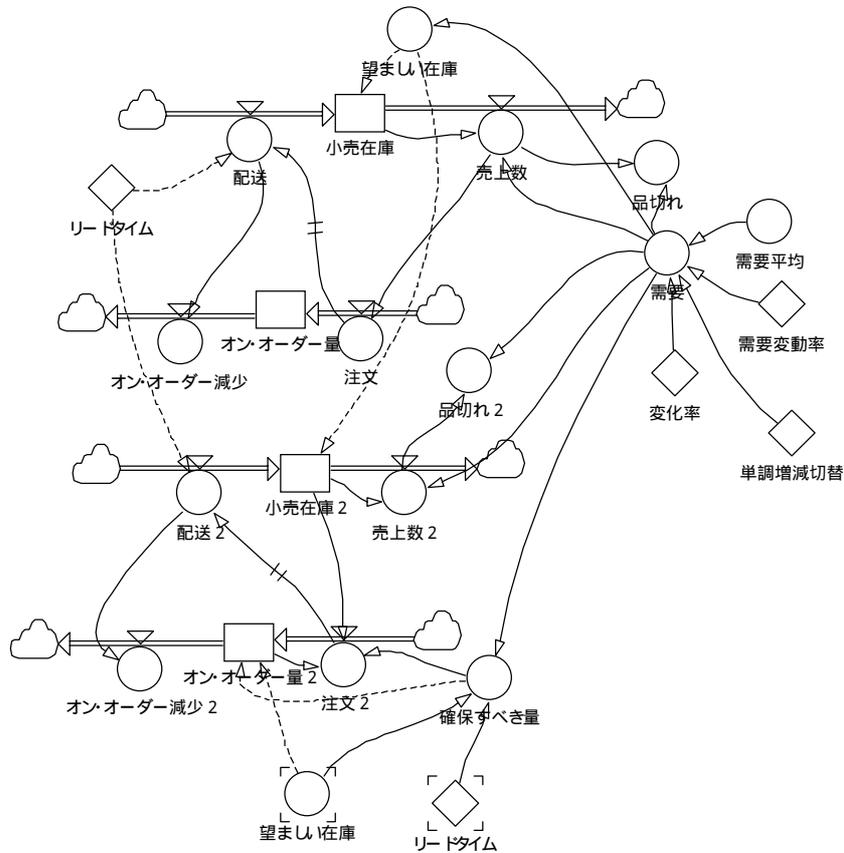
モデルの意図

例 5 - 3 小売店在庫維持モデル 1 のモデルで，発注の仕方を変化させて店舗の理想在庫維持政策についての意味合いを探ることが目的である。操業条件は，例 5 - 3 と同じであるが，発注方法が異なる。

モデルの構造

モデルの構造および操業条件は以下の通りである。

店舗の「リードタイム」と発注間隔（1日）を考慮して，それらの和の期間中に必要な量を確保するように発注する。すなわち，その期間中に発生する需要量に，理想店舗在庫，すなわち，「望ましい在庫」40着を加えた量を「確保すべき量」と定義し，その量を店舗が確保することを計画している。「注文2」は，その「確保すべき量」と，現在の確保量（「小売在庫2」と「オン・オーダー量2」の和）の差を埋めるように決める。売れた量（「売上数2」）を発注するわけではない。



発注方式が異なる主体と例 5 - 3 の主体の両方を同時に構造化して比較できるようにしている。変数の末尾に 2 がついたものが上記の新しい発注方式を採用する主体の変数である。

る。需要パターンは先の4種類を例5 - 3の説明に基づいて実行できる。新しい主体2では「確保すべき量」が付加され、その量から「小売在庫2」と「オン・オーダー量2」の和を引いた量がプラスの場合にはそのプラス量だけ発注し、マイナスの場合（確保すべき量より保有量が多い）には発注しないという行動になる。

シミュレーション演習のポイント

売れた量を注文する場合と、確保すべき量に足りない分だけを注文する方式の違いがいかなる場合に生まれるかを確認する。前者の注文方法では品切れ量は注文されないのに対して、後者の注文方法では品切れした場合には確保すべき量だけ注文することになるので、確保すべき量の水準まで戻すような注文をするという違いがある。品切れが発生しない場合には、売れた量と、確保すべき量を下回った分は等しいことになることも理解する。またリードタイムを変更して発注方法の違いに与える効果も興味深い。

第6章のモデル解説

199頁 例6-2 プッシュ方式モデル

モデルの意図

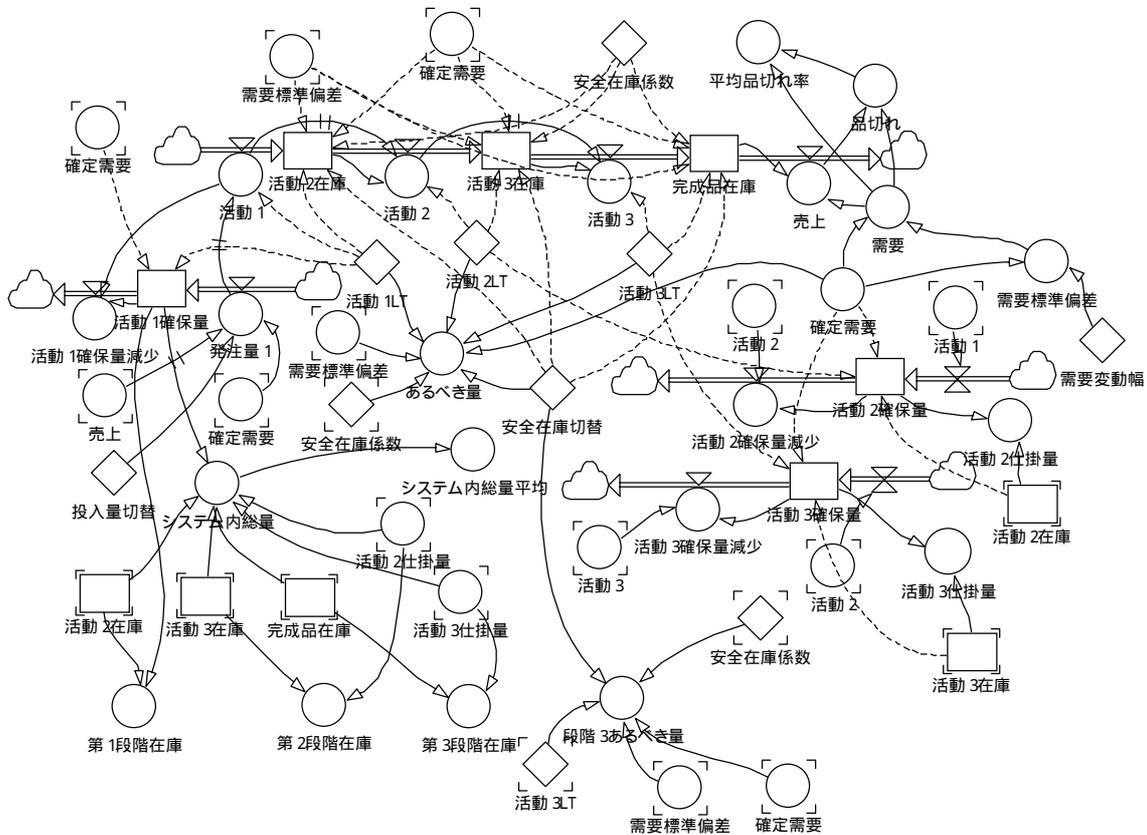
いくつかの活動が縦列したチェーン構造におけるプッシュ方式の基本特性を理解することが目的。

モデルの構造

モデル構造は下のようになっている。操業条件は以下のようになっている。

「需要」は一定量 100 個という場合と、正規分布状に変化する場合の 2 通りを想定する。チェーンの最初の活動 1 から「発注量 1」として平均需要分（「確定需要」）を翌日にチェーンに流し込む。

活動の能力については制約がないとする。各活動は終了したものをその後の活動にそれぞれのリードタイム（「活動 1 LT」、「活動 2 LT」、「活動 3 LT」）だけ遅れて流す。



物の流れは左から右への方向である。「あるべき量」はチェーン内に確保しておくべき量のことである。「システム内総量」はチェーン内に存在する在庫（「活動2在庫」、「活動3在庫」、「完成品在庫」）と仕掛在庫（「活動2仕掛量」、「活動3仕掛量」、「活動1確保量」）の総和で現在、チェーンに確保している量である。また、「第1段階在庫」は「活動1確保量」と「活動2在庫」の和で、活動1において確保している量になる。同様に、「第2段階在庫」は、「活動2仕掛量」と「活動3在庫」の和で活動2段階において確保している量になる。そして、「第3段階在庫」は「活動3仕掛量」と「完成品在庫」の和で最終段階の活動3段階で確保する量である。

「活動2確保量」は活動1を終了してから活動2を終了するまでのプロセスにある量で、活動1後の在庫である「活動2在庫」と、「活動2仕掛在庫」の和になる。同様に、「活動3確保量」は、活動2を終えて、活動3を終えるまでのプロセスにある量になる。それは活動2後の在庫である「活動3在庫」と、「活動3仕掛在庫」の和である。しかしながら、それら「活動2確保量」と「活動3確保量」は命名が紛らわしいが、前出の「第2段階在庫」と「第3段階在庫」とはそれぞれ異なることに注意すること。チェーンの構造はたいして複雑ではないが、実際の動きで数字を適切に処理するにはこのような煩雑さが加わるのは仕方がない。

操作の基本

需要パターンを変化させることを説明する。「需要」をダブルクリックし、「需要」Auxiliary Symbol of Properties 画面にする。以下の式が Definition 欄に出ている。

(確定需要+NORMAL(0, 需要標準偏差,.9897))

この定義を見るとわかるけれども、以下のようにして需要のバリエーションを生成できる。

需要の変動幅は、「需要標準偏差」をダブルクリックし、「需要標準偏差」Auxiliary Symbol of Properties 画面とし、その Definition 欄に定義されている下のステートメントの 0.2 を希望の値、例えば 0.3 に英数半文字で書き換える。そうすると需要の変動（標準偏差）は 30 になる。すなわち、

0.2*確定需要

を **0.3*確定需要** とする。

なお、上の正規分布の規定を行っている NORMAL 関数は正規分布の乱数を発生させる組み込み関数で、以下のように定義する。

NORMAL (平均値, 標準偏差, 乱数発生のための数)

例では標準偏差は「確定需要」(100)に0と1の間で定義された「需要撓乱」がかかって規定されるようになっている。乱数発生のための数は0未満で定義する。これを定義するとシミュレーションにおいて出てくる乱数が固定化される。定義しなければ、すなわち、NORMAL(平均値, 標準偏差)だけであれば、シミュレーションごとに異なる乱数が発生する。

活動1から流し込む量を変化させることを考える。「発注量1」をダブルクリックし、「発注量1」Auxiliary Symbol of Properties 画面にする。そうすると、Definition 欄では

確定需要

となっている。もし、100個よりももっと流し込みたい場合には、例えば103個を流し込みたいとすると、それを**確定需要+3**とすれば良い。

シミュレーション演習のポイント

この例ではプッシュ方式の特性を理解する。各在庫の水準はリードタイムを変えても変化しないが、チェーン内に「**あるべき量**」と「**システム内総量**」はリードタイムの変化に応じて増減することに注意。また、需要が変動すると品切れが発生するが、それをなくするために活動1から流し込む量を増やすといかなるチェーンになるかも検討する意味がある。

モデルの意図

例6-2と同じチェーンで売上量を前の主体に注文するいわゆるプル方式の運営をする場合の結果を確認しながらプル方式の特徴を理解することが目的である。例6-2と異なる点は各活動主体の発注方法である。

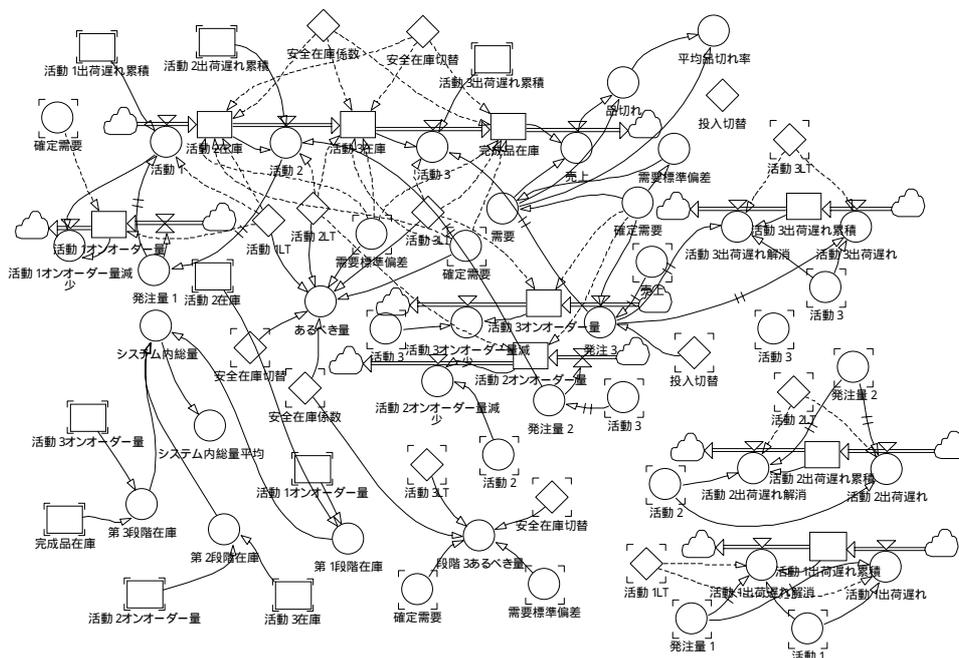
モデルの構造

以下が例6-3のモデル構造である。

活動3の主体はその日に売れた量(「売上」)だけ供給しようとする。しかしながら、実際に供給できる量(「活動3」)は活動3の前にある在庫の量(「活動3在庫」)しか処理できないので、売れた量の注文である「発注3」と「活動3在庫」の少ない方の水準が完成品在庫への供給量(「活動3」)になる。そして実際に供給した量(「活動3」)を前の活動2へその日の営業終了後に注文する(「発注2」)。その供給は活動3のリードタイム(「活動3LT」)分だけ遅れる。活動2も同じように行動する。活動1はその前に在庫がなく、活動1で注文した量は活動1のリードタイム(「活動1LT」)だけ遅れて入手できる。

活動2、活動3の供給量(「活動2」と「活動3」)はそれらの前にある在庫量(「活動2在庫」と「活動3在庫」)と受けた注文量(「発注3」と「発注2」)のうち少ない方で決まる。それゆえ、受けた注文量に応じ切れない品切れも発生しうる。その場合には次の供給で品切れを補充する行動を想定する。需要に応じきれない完成品の「品切れ」については後から補充できない。お客は待たないことを想定しているからである。

その他は例6-2と同じである。



注文情報は需要に近い側から活動 1 方向に流れる。それら注文に対して各活動のリードタイムだけ遅れて供給される。

品切れをなくすために、各在庫の初期在庫を増やすことを考える場合には、**安全在庫切**をダブルクリックして **1** とする。安全在庫そのものを増やしたい場合には、**安全在庫係数**をダブルクリックし、デフォルトで入っている **2.33**（許容品切れ率 1% の場合の安全係数となっている）を、例えば、**3.0** などのように大きくする。

需要の変動を変える場合には、**需要標準偏差** をダブルクリックし、**‘需要標準偏差’ Auxiliary Symbol of Properties** 画面にする **0*確定需要の 0** を、例えば、**0.2** と置き換える。これは確定需要量の 20% にあたる変動を意味する。変動幅を大きくしたい場合には **0.2** を 1 に近づける。これらの値の変更方法は今までの手順と同様である。

「**システム内総量平均**」は「**システム内総量**」のシミュレーション期間中の平均値である。「**平均品切れ率**」は「**品切れ**」のそれである。これらはモデルに組み込まれた組み込み関数を用いて計算されている。例えば、「**平均品切れ率**」をダブルクリックすると、**‘平均品切れ率’ Auxiliary Symbol of Properties** 画面となるが、その **Definition** 欄では、

RUNAVERAGE (品切れ/需要)

となっている。**RUNAVERAGE** は括弧内の計算値（「**品切れ**」を「**需要**」で割った値）をシミュレーション期間中で平均する組み込み関数である。

シミュレーション演習のポイント

プル方式の特徴を理解する。まず第 1 に、リードタイムの影響がチェーン内総量に影響することはもちろん、各在庫における変化をもたらすことを確認する。チェーン内を流れる量が例 6 - 3 のプッシュ方式と異なってチェーン内に取り込まれる量が一定ではない。売上量に応じて変化する。リードタイムの変化は在庫から流出する量および流入する量のパターンを変えるので各在庫の動きも変化してくる。

第 2 に、需要変動に対してプル方式がいかなる反応をするかを理解する。品切れの発生にも注意する。品切れを例 6 - 3 と同じ程度にするにはどれだけ増やすべきか、そしてチェーン内および各在庫のその時の水準がどうなるかを理解する。プル方式にしたから在庫が当然減ると短絡的に考えることの危険性を理解する。プル方式は品切れに対する対策が焦点になることが鍵である。

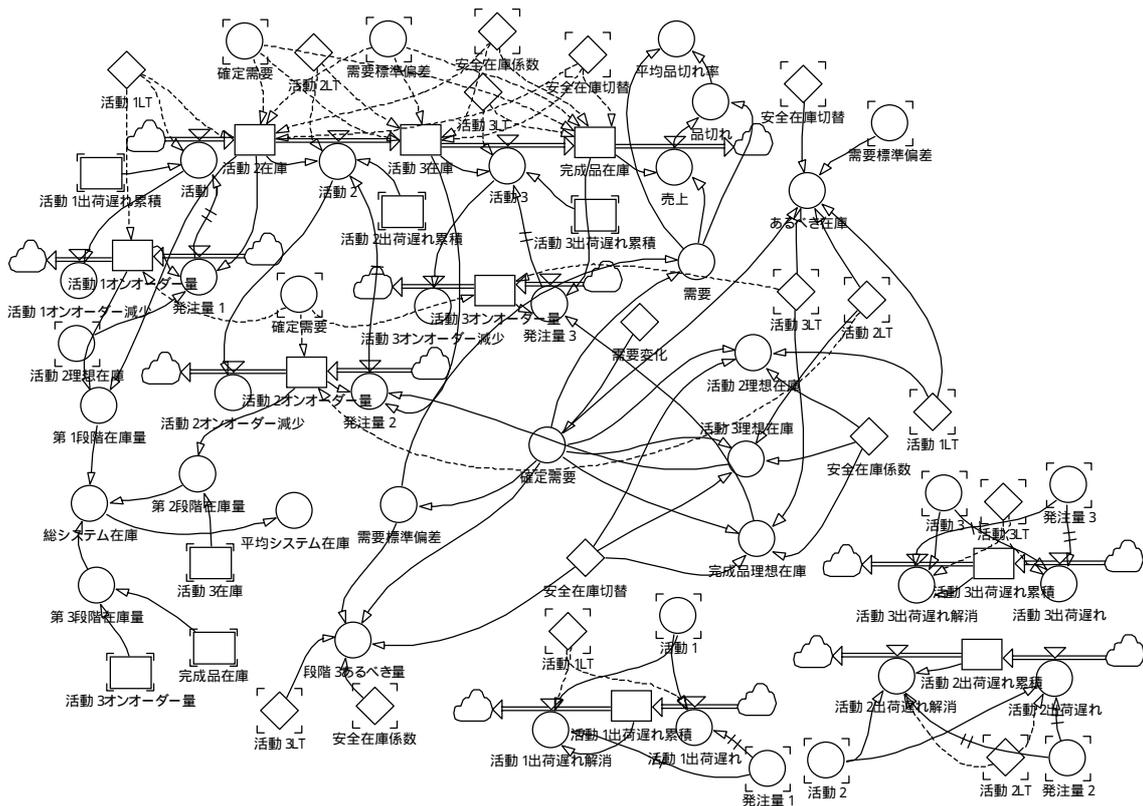
モデルの意図

チェーン全体における適正在庫水準という概念を理解し，それを常に維持するようにチェーン内の発注を行う場合，各活動主体がそれぞれのリードタイムに応じてまずそれ自身の適正在庫（「活動2理想在庫」，「活動3理想在庫」，「完成品理想在庫」）を規定し，それを維持するように発注を行うという方法を理解する。

モデルの構造

下の図がモデルの構造である。操業条件は，以下の通りである。

各主体は自分自身の課業（生産加工など）にかかる供給リードタイムに発注間隔を加えた総リードタイム中の需要に相当する量を適正在庫とみなし，それを維持するように発注する。例えば，活動主体2は「活動3在庫」（活動2が終了したものの在庫）と活動主体2の仕掛在庫（「活動2オン・オーダー量」）の和が「活動3理想在庫」に等しくなるように「発注量2」を決める。それは活動主体2の供給指示量になるが，実際に「活動3在庫」（活動主体3が活動できる状態になっている量）に入ってくるのは活動2のリードタイム（「活動2LT」）後である。活動2にとりかかる時にはその前の在庫，すなわち活動2在庫にその活動ができるだけの在庫がなければならない。それは活動主体1が確保しておくことになる。活動主体1，3も同じように行動する。各活動量はその前にある在庫量以上にはならない。足らなかった分は後で補充される。



適正在庫は2種類ある。まず個々の活動主体に関するもので、それらは「活動1理想在庫」、「活動2理想在庫」、「活動3理想在庫」である。2番目の適正在庫はチェーン全体に関する適正在庫で、「あるべき在庫」として定義されている。

適正在庫および3つの理想在庫に安全在庫を加えるかどうかは「安全在庫切替」で決める。それを1とすると加算され、0とすると加算されない。その場合の「安全在庫係数」はデフォルトで2.66となっている。

この独立方式モデルでは需要の変化についてバリエーションを増やしている。それは後でプッシュ方式とプル方式との差異を示すためである。そのバリエーションは、「確定需要」をダブルクリックして開くとわかる。開いた「確定需要」Auxiliary Symbol of Properties画面において、その定義は以下のようになっている。

100+SINWAVE(30*需要変化,60,0)

すなわち、「需要変化」が1の場合には平均需要100個がプラス、マイナス方向に、ピークからピークまで60日間かけて30個振れる。これは標準偏差の変動ではない。「需要変化」が1の場合には平均需要100個は不変である。「需要変化」を変える場合には、それをダブルクリックして開いた「需要変化」Auxiliary Symbol of Properties画面において、Definitionで0とすると平均値の変化はなく、1とすると上述の周期変化をする。例6-5においてプッシュ方式の適正確保量の不応モデルを動かす場合に、需要に周期変動を入れたサイン曲線を使うが、それに対応した状況で独立方式を動かす場合には、「需要変化」を1とする。

シミュレーション演習のポイント

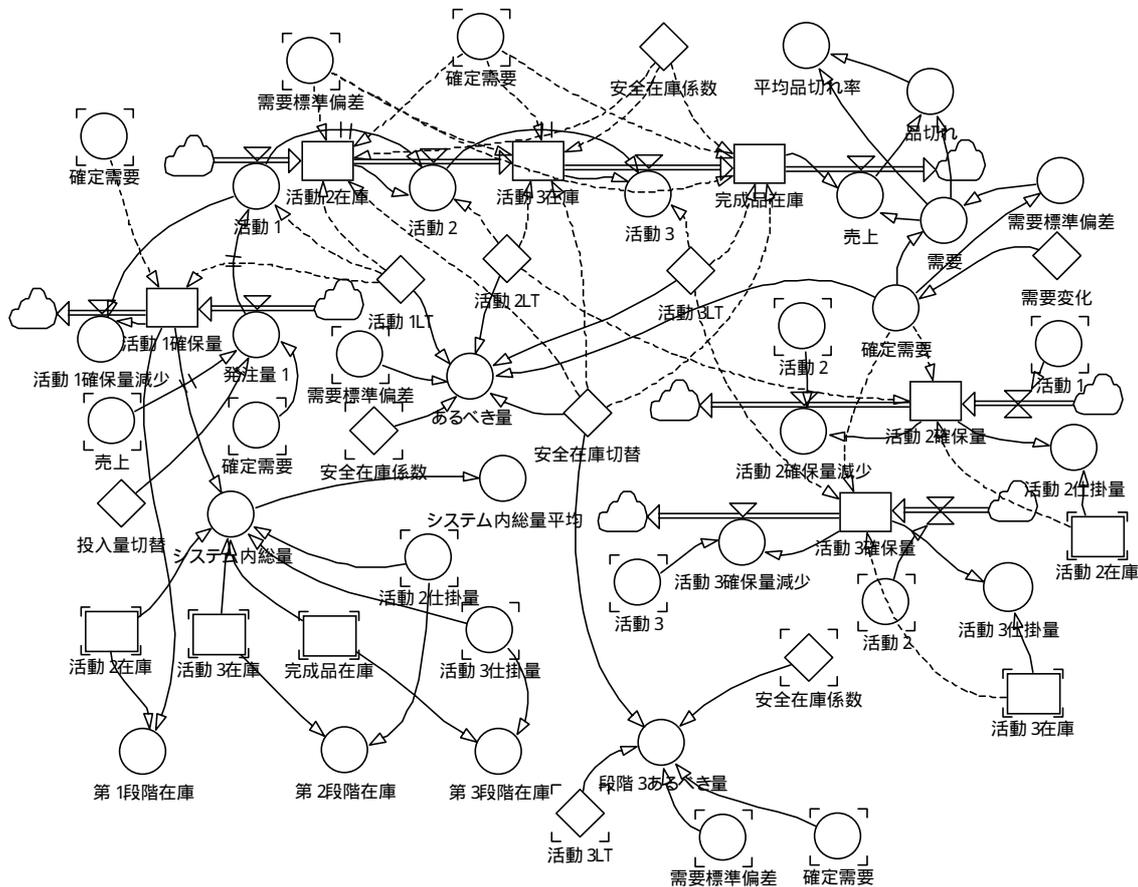
標準的なチェーンの運営方式を理解する。各主体が独立に自身のリードタイムを考慮しながら適性在庫を保持する行動が基本にあるのでそれぞれの在庫の変動はあるが、制御ができる。リードタイムの変化、需要の変化に対応して適性在庫を調整することで制御力が堅持できることを理解する。例6-2のプッシュ方式および例6-3のプル方式において、例えば品切れを回避する方法と比較すると良い。

モデルの意図

需要が構造的に変化する場合のプッシュ方式の特性を理解することが目的。サプライチェーンの運営ではプッシュ、プルという概念よりも適正在庫の維持ということの方が重要であることを理解してもらう。作業条件は需要を除いて例6 - 2と変わらない。

モデルの構造

需要の変化を入れた構造でそれ以外は例6 - 2のプッシュ方式モデルと変わらない。この変化については独立方式モデルの「**需要変化**」についてと同様であるのでそれを参照のこと。



シミュレーション演習のポイント

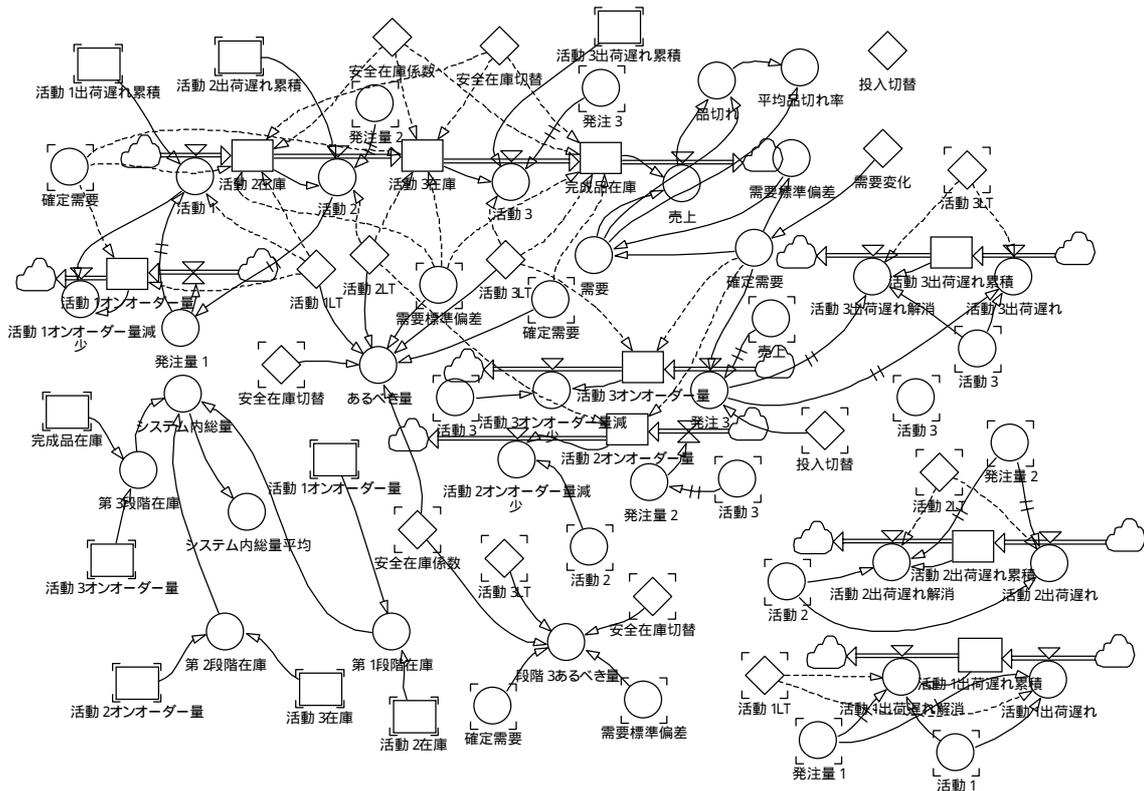
プッシュ方式は流し込む際にチェーン全体の適正確保量という概念を考慮しない。従って、需要の構造は売れた量だけで判断され、それが増えれば流し込む量も増え、逆に減ると流す量も減る。しかし、需要自体の構造については考えていないのでどうしても対応は遅れる。需要の減少局面、増加局面に対する反応に注意して欲しい。

モデルの意図

プル方式の欠点を需要構造変化への適応特性から学ぶ。条件は例6-5のプッシュ方式と同じである。ただし、今度は活動3から引いていく。

モデルの構造

このモデルの構造は以下の通りである。新たな部分は、需要変化の部分だけである。これの変化については独立方式モデルの「**需要変化**」についてと同様であるのでそれを参照のこと。



シミュレーション演習のポイント

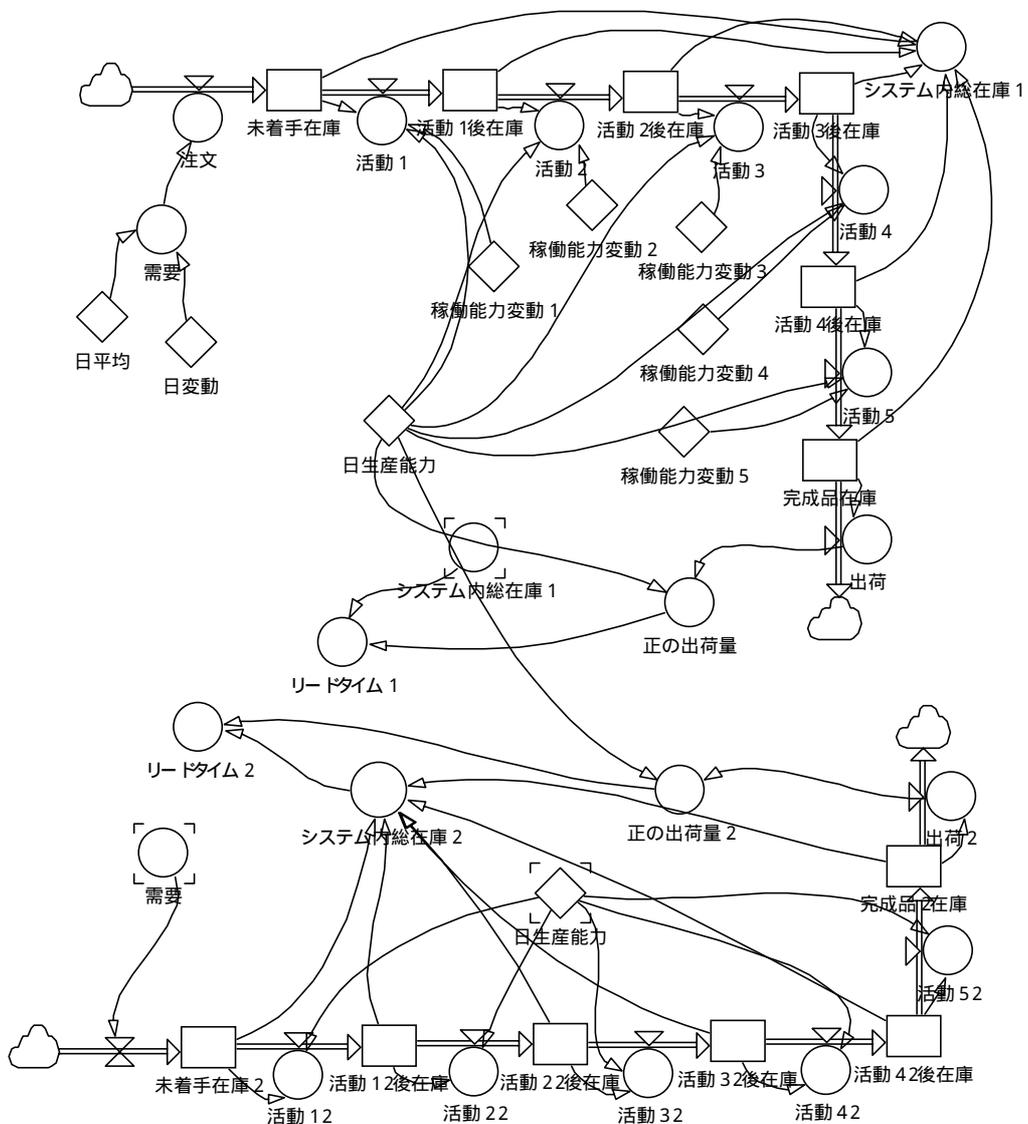
プル方式は引っ張る際にチェーン全体の適正確保量という概念を考慮しない。従って、需要の構造は売れた量だけで判断され、それが増えれば引く量も増え、逆に減ると引く量も減る。しかし、需要自体の構造については考えていないのでどうしても対応は遅れる。品切れが発生しても気にしないで、需要は売れた量だけあったと想定している。需要の減少局面、増加局面に対する反応に注意して欲しい。

モデルの意図

リードタイムはいままで、既定のように扱ってきた。少なくとも最初にチェーンを設計する場合には標準的なリードタイム(推定のリードタイム)を用いる。しかしながら、現実にはリードタイムは変化するもので、そのことを理解してもらう。特にボトルネック現象が発生することで推定していたリードタイムとは違うリードタイムになることを理解する。

モデルの構造

本モデルでは2種類のプロセスが比較できるように一緒に含まれている。



上の変数に1桁の添え字がついたプロセスは活動能力の変動がある場合で、下の2桁の添え字のあるプロセスは稼働能力がそのまま変動なしに発揮できる場合のプロセスである。

同じ需要パターンに対してこれら 2 つのプロセスが動いて完成品を供給するかを分析する。

モデルで各活動の稼働能力が変動する程度は菱形の「稼働能力変動 i ($i=1,2,3,4,5$)」をダブルクリックして変更することができる。例えば、「稼働能力変動 1」をダブルクリックすると「稼働能力変動 1」Auxiliary Properties の Definition 欄の数字 0.1 を半角英数字で書き換えることで可能である。また「需要」は平均 10 個，標準偏差 2 個の正規分布となっている。

シミュレーション演習のポイント

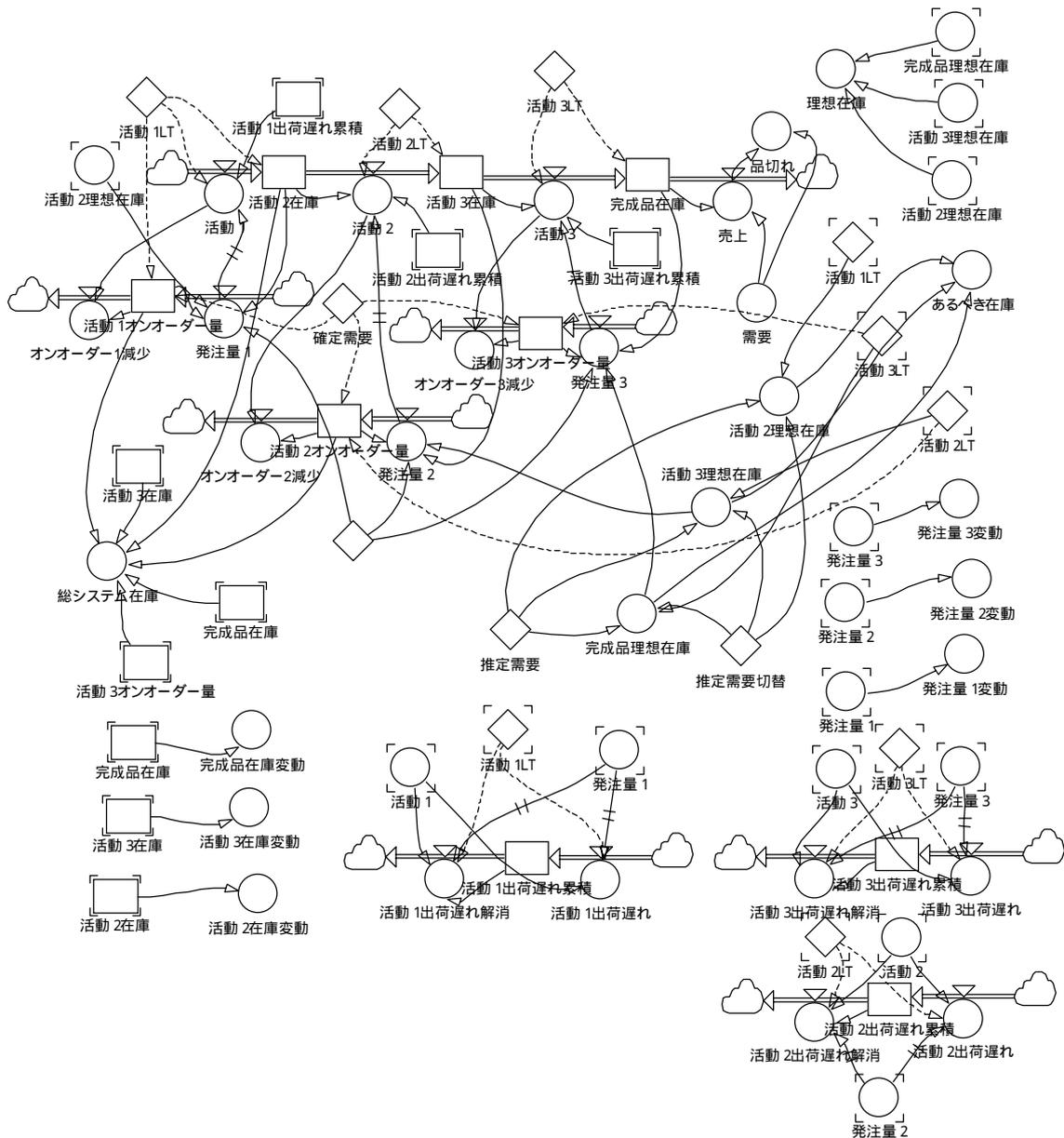
活動がいくつもつながっているのが現実のチェーンであるが，それら活動がすべて予定通りに稼働すると期待する方が難しい。その予定からの乖離がいかなるリードタイムを帰結するかを確認する。このモデルではリードタイムはすべて 1 日ということになっているのでオン・オーダー量というものは無い。各活動の稼働能力変動を変化させてリードタイムの変化を理解する。その場合に，与える変動と在庫の変化，そしてリードタイムの変化の関係を理解することも重要である。

モデルの意図

ブルウィップ効果を独立方式のモデルで確認する。特に，発注量を決める場合に，オン・オーダー量を考慮することを忘れてと，とんでもない事態に陥ることを理解することが目的である。製品がヒットしたりするとこの種の現象は現実にも頻繁に起こる。適切な発注方式を堅持することも重要性を理解してもらう。

モデルの構造

モデルの構造は例6-4 独立方式モデルと同じであるが，ブルウィップ効果を生じさせるためのメカニズムが組み込まれている。



第1は、「 α 」というオン・オーダー量を考慮する程度を示す係数、第2は需要の推定誤りを導入する「**推定需要**」の導入である。「 α 」はそれを1とすると適切な発注を意味し、1より逸脱すると不適切さが増すことになる。起こりやすいのは1を下回ること、これは今までに発注したものが少な過ぎるか、あるいは忘却してしまうことを意味する。

また「**推定需要**」は適切な水準は100個であるが、それを低く見積もり過ぎてしまう場合や高く見積もり過ぎるような場合に使う。これを100個以外の数字にする場合には、同時に「**推定需要切替**」も1にしなければならない。「**推定需要切替**」が0と置かれている場合には、「**推定需要**」は効かない。

シミュレーション演習のポイント

不適切な発注をすることがチェーンのパフォーマンスにいかなる影響を与えるかを理解することがポイントである。「 α 」あるいは「**推定需要**」の変更を試みることでそれは可能である。またブルウィップ効果はリードタイムによってもその発現が一層影響されることも試みる価値がある。

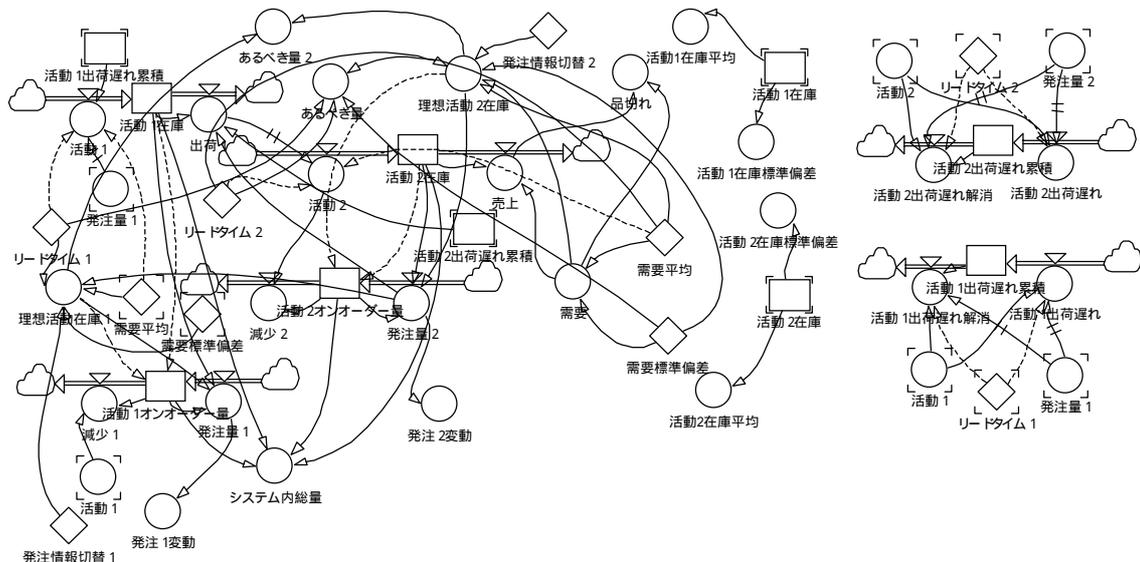
モデルの意図

活動主体がその発注を決める時に基準になる適正水準がいかに決められるかによってプルウィップ効果も影響されることを理解することが目的になる。チェーンを構成する 2 人の主体がそれぞれの適正在庫水準を決める需要情報として何を使うかがその場合に重要である。

プルウィップ効果そのものは発注主体が適正と考える需要水準を見定め、それを首尾一貫して採用することによって適正な発注方式を堅持することで回避できることを理解する。なお、2 人の主体の発注行動ではそれぞれが独立発注方式を用いている。

モデルの構造

モデルは 2 段階のチェーンとしてより単純化している。デフォルトでは両主体が市場需要の平均、標準偏差という特性値をベースに適正確保量を堅持するように発注行動することが想定されている。発注行動を変化させるには、主体 1 に関しては「発注情報切替 1」、主体 2 に関しては「発注情報切替 2」を変化させる。それらを 1 とすると正確な需要の特性値を用いて適正確保量（「理想活動 1 在庫」および「理想活動 2 在庫」）を設定するが、0 にするとすぐ後の主体の発注情報そのものをそれらの設定で用いることになる。



シミュレーション演習のポイント

適正確保量に用いる需要情報の重要性を理解することである。同時にリードタイムの長さもそれに影響することも理解する。

第7章のモデル解説

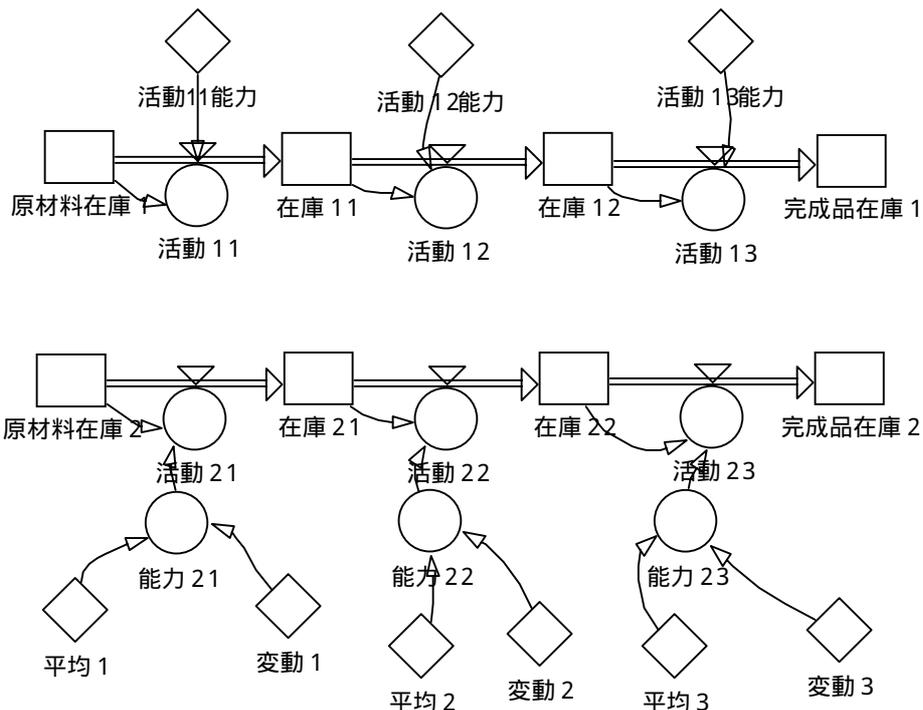
260頁 例7-1 ボトルネック・モデル1

モデルの意図

能力制約が活動（例えば生産活動）に課される場合には，それらがチェーンにおける供給量（通過量）を規制することで所要供給量を満たすためのリードタイムを左右するという現象を理解することが目的である。

モデルの構造

モデルの構造は以下のようになっている。



2つの工程があって，それらと比較するようになっている。原材料に1,000個の完成品を作ることができる原材料があるとする。後は3つの活動を経て，1,000個が完成するまでにどれだけ時間がかかるかを計算する。下の2桁の数字を添えている活動では，能力稼働に関する制約がある。それら能力は確率的に変動する可能性がある。上の工程にも能力制約はあるが，変動はしない。

能力変動は正規分布で発生する。例えば「能力21」では，それをダブルクリックすることで「能力21」Auxiliary Properties画面になる。そのDefinition欄には以下のような記述がある。

INTEGER(NORMAL(平均 1,変動 1, .9898))

「平均 1」,「変動 1」を変えることで分布形状が変わる。それらを変えるには, 菱形の「平均 1」,「変動 1」をダブルクリックし,「変動 1」Auxiliary Properties 画面または「変動 1」Auxiliary Properties 画面にしてそれらの Definition 欄の数字を変えることが必要になる。上の「能力 2 1」の定義式で, INTEGER は () 内の数字を整数化する組み込み関数である。NORMAL(平均 1,変動 1, .9898) から出てきた数字は小数の可能性があるが, それを整数にするわけである。現実的な状況にしてより正確にリードタイムを測るためである。

シミュレーション演習のポイント

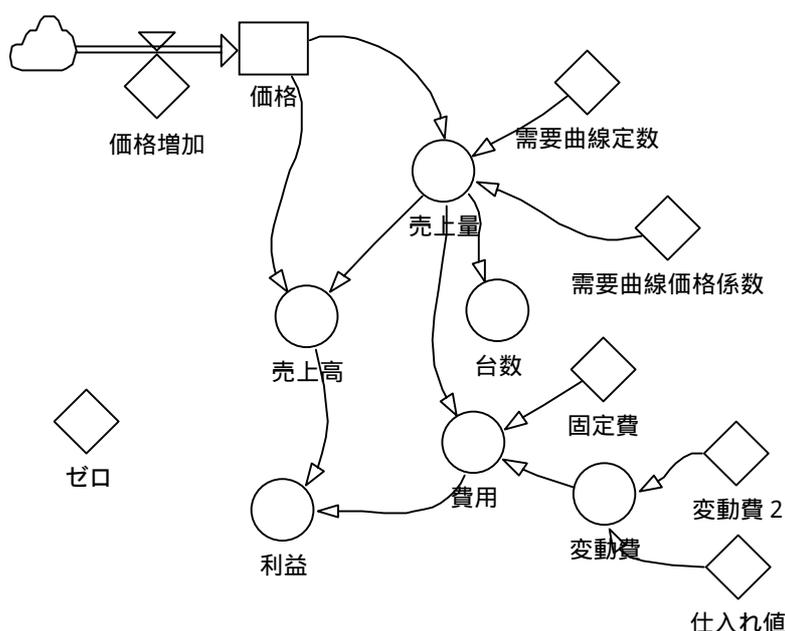
能力制約のばらつきを変えることでどこに在庫が滞留し, リードタイムにどう影響するかを確認する。在庫滞留がリードタイムを長くすることを理解する。

モデルの意図

価格をいろいろと変化させた場合の利益を計算するのにモデルを使う。今までのシステムの動きを計算する考え方と若干異なる使い方である。モデルの利用方法について別の形態を理解してもらう。

モデルの構造

モデル構造は下のようになっている。



「価格」はある水準から「価格増加」の刻みで高くしていくようにして決まる。スタート価格は価格をダブルクリックして、「価格」Auxiliary Properties 画面にし、Definition 欄にある数字である。現在は 0 である。刻みは「価格増加」をダブルクリックし、Definition 欄にある数字になっている。現在は、0.1 になっている。現在、モデルは 1,200 期動くが、それは $120(0.1 \times 1200)$ 円までの価格の状況を作り出す。110 円以上になると需要は 0 になるようになっている。

「売上量」の定義は、それをダブルクリックして「売上量」Auxiliary Properties 画面にするとわかる。Definition における定義は以下のようになっている。

IF(価格 110,0,需要曲線定数-需要曲線価格係数*価格)

すなわち、「価格」が 110 円を超えると「売上量」は 0 で、それ以下では**需要曲線定数-需要曲線価格係数*価格**という需要関数で決まる。IF は条件文で論理演算を指示する。ここを変える場合には、例えば、120 円を超えたら 0、それ以下では需要関数通りとした場合には、**IF(価格 120,0,需要曲線定数-需要曲線価格係数*価格)**となる。

需要関数は「**需要曲線定数**」,「**需要曲線価格係数**」,「**価格**」の関数になっている。「**価格**」以外はモデルでは一定値で与える。菱形の「**需要曲線定数**」,「**需要曲線価格係数**」をそれぞれダブルクリックしてそれらの値を定義する。現在は,それらはそれぞれ **210,000** と **2,000** になっている。いい加減な値を入れると利益はとんでもない動きになるので注意すること。

「**利益**」は「**固定費**」と「**変動費**」を「**売上高**」から減じて計算されるようになっている。「**固定費**」は一定の **200,000** 円になっている。「**変動費**」は「**仕入れ値**」によって決まる部分と,ステーションの稼動にともなう1リットルあたりの操業変動費である「**変動費 2**」が加算されたものになっている。それらの和の「**変動費**」が「**売上量**」に掛かって操業総変動費が計算される。単位あたり操業変動費「**変動費 2**」は **3** 円になっている。

シミュレーション演習のポイント

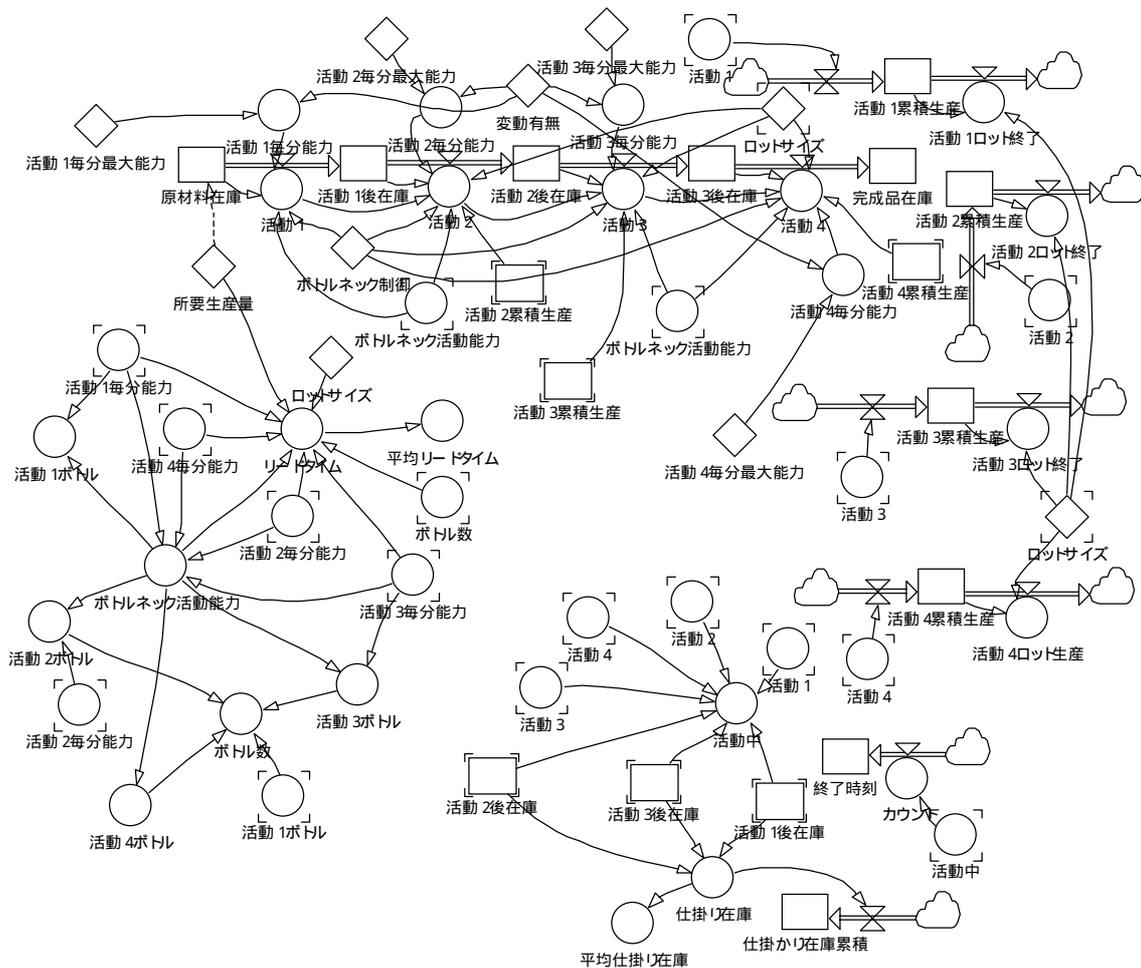
費用や需要関数を変化させて最大利益の売上量がどのように変化するか,それによってボトルネックの評価額がどのようになるかを理解する。しかしながら,費用や需要関数を変化させるときには注意すること。

モデルの意図

本モデルでは4つのつながった活動が500個の生産量をこなしていくプロセスを考え、そこでリードタイムがどのように決まるかを理解する。ロットサイズ、ボトルネックの両要素の影響を考えることが目的である。まず活動1を終え、次に活動2へと流されていく。最後は活動4で処理して完成である。

モデルの構造

活動にはリードタイムはあえて規定せず、分当たりの処理能力(「活動1毎分能力」、「活動2毎分能力」、「活動3毎分能力」、「活動4毎分能力」)だけを規定している。「原材料在庫」から500個分の量が工程に流し込まれ、4つの活動を経て完成品として「完成品在庫」に累積されてくるまでの工程における在庫の動きをフォローし、最終的な所要時間(「終了時間」)がいかにか決まるかを計算するモデル構造になっている。本文の図7-6では横軸は1目盛りが20時間になっている。



「**リードタイム**」は理論式による計算式である。工程を流すロットの大きさは「**ロット**」によって変更できる。「**ロット**」をダブルクリックし、**Definition** に 1 から 500 の間の数字を入れる。デフォルト値は **1** になっている。またボトルネックに合わせた量を流すような制御方法も「**ボトルネック制御**」を 1 とすることで採用できる。デフォルトは **0** になっている。

なお、ロットを大きな値にする場合には、まず図の横軸を大きくしなければならない。現在の図では 500 個をすべて完成するまでには至らないこともある。それはフローが出ている画面で、上の **Simulation** をクリックして開き、一番下にある **Simulation Settings** を一回クリックして開いた画面で一番左にある **Simulation** を開いて **Stop Time** の現在のデフォルト値 **1:00:00** を例えば **3:00:00** とする。それによって 3 倍までシミュレーション期間を延ばすことになる。またシミュレーション結果の図、すなわち本文の図 7-6 で縦軸を延ばすには、図のコーナーないしエッジをダブルクリックして、**Time Graph Control Properties** 画面を開く。そこで **Value Axis** をさらに開く。その画面で左下の **Scale** のところで **Minimum** と **Maximum** のところをクリックしてその右側の四角の欄に希望の限界値を打ち込む。

シミュレーション演習のポイント

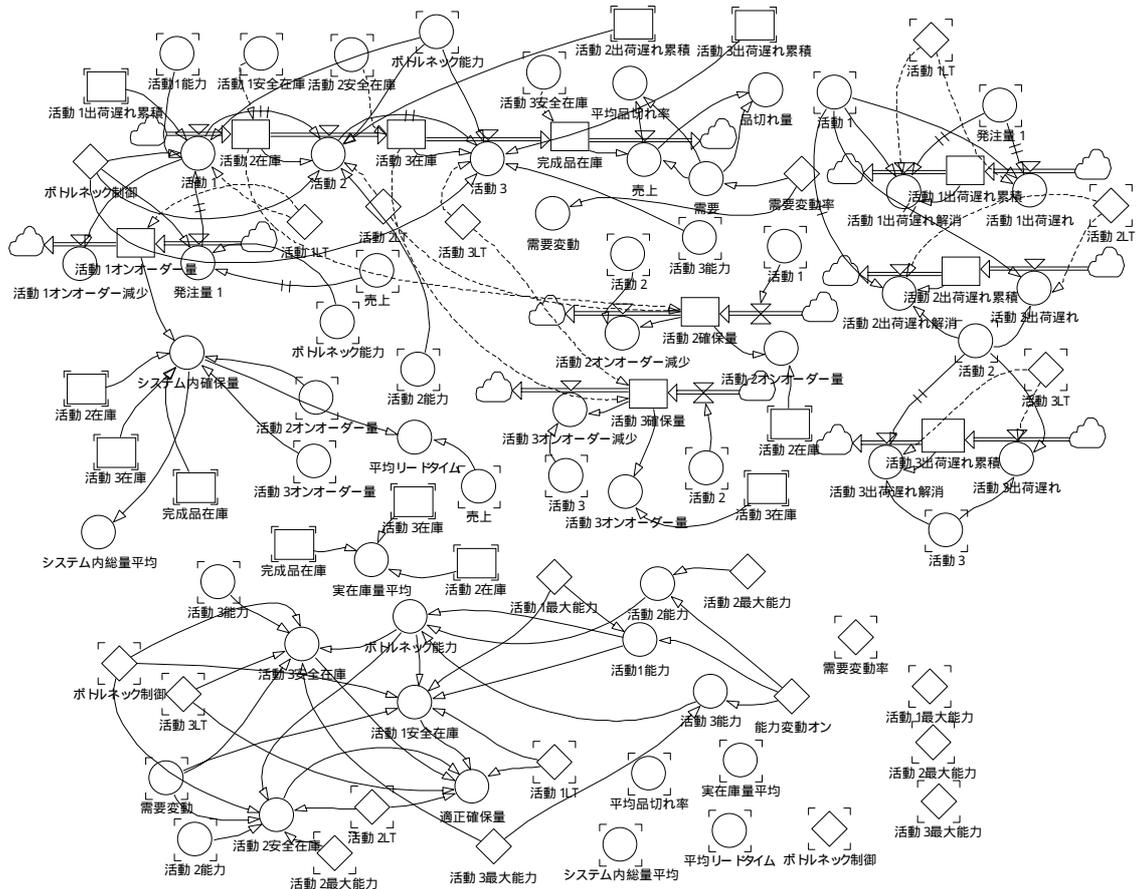
能力の変動、流すロットサイズの大きさなどを変えながらリードタイムの変化を調べる。それによってそれらの間関係についての感触を得る。

モデルの意図

プッシュ方式で流しているチェーンにおいてボトルネックがいかなる影響を与えるかを確認し、それによってプッシュ方式の特性を理解することが目的である。

モデルの構造

3つの活動がつながっているチェーンで、それぞれの活動の事前にわかっているリードタイムがある。このリードタイムはそれぞれの活動の能力を前提にそれらがフル稼働した場合に予想できるリードタイムとして入れられている。



モデルは若干複雑になっている。まず「発注量1」によって売れた量（「売上」）をチェーンの左から流し込む。「活動1」は「発注量1」を「活動1LT」（デフォルト値は8日）だけ遅れて終了する。しかしそれは1日当たり「活動1最大能力」以内でなければならない。「活動1最大能力」は現在、デフォルト値として200個が入っている。他活動の2倍の能力になっている。「活動2」は「活動1」から供給された量を翌日から引き継いで1

日当たりの能力の範囲内でその活動（加工とかその他の供給活動）を行う。能力範囲内では「活動 2 LT」だけかかって終わった量は翌日に次の「活動 3」へと送り出す。能力を超えてできなかった分は翌日になるので、実質のリードタイムは長くなる。「活動 3」は「活動 2」を引き金にその活動を開始し、能力範囲内のものは「活動 3 LT」だけかかって完成品在庫へと送り込む。

ボトルネックに合わせた流し方をする場合には、菱形の「ボトルネック制御」の値を 1 とする。0 はその制御はしないことを意味する。

需要は下のような定義式になっている。正規分布とサイン関数の半々の合成である。

**INTEGER(MAX(0,(100+NORMAL(0,需要変動率*100,0.9897))*0.5
+0.5*(100+SINWAVE(100*需要変動率,90,0))))**

「需要変動率」を変化させると需要の変動が大きくなる。その「需要変動率」のデフォルト値は 0.2 になっている。正規分布（サイン関数）だけにしたければ上式の正規分布に係数として付いている 0.5 を 1（0）に、サイン関数に付いている 0.5 を 0（1）にする。

本文に図として出てくるさまざまな結果は、菱形の「活動 1 最大能力」、「活動 2 最大能力」、「活動 3 最大能力」、さらに「能力変動オン」、「ボトルネック制御オン」の数字を変えることによって可能である。それぞれをダブルクリックし、それぞれの **Auxiliary Properties** 画面の **Definition** 欄における数字を変える。「能力変動オン」、「ボトルネック制御オン」は制御する場合には 1、しない場合には 0 である。

チェーンの稼働のパフォーマンスは、「平均品切れ率」、「平均リードタイム」、「システム内総量平均」、「実在庫量平均」を見る。シミュレーションが終了した段階で、それらのパフォーマンス尺度にカーソルを置くと結果の数字が出てくるようになっている。

シミュレーション演習のポイント

演習ではさまざまな制御および能力変動、さらにはリードタイムなどを変化させながら結果の変化を見ることでチェーンの挙動がそれらによってどう影響されるかを考える。またボトルネック現象に対応するプッシュ方式の特性としてそれらを理解する。

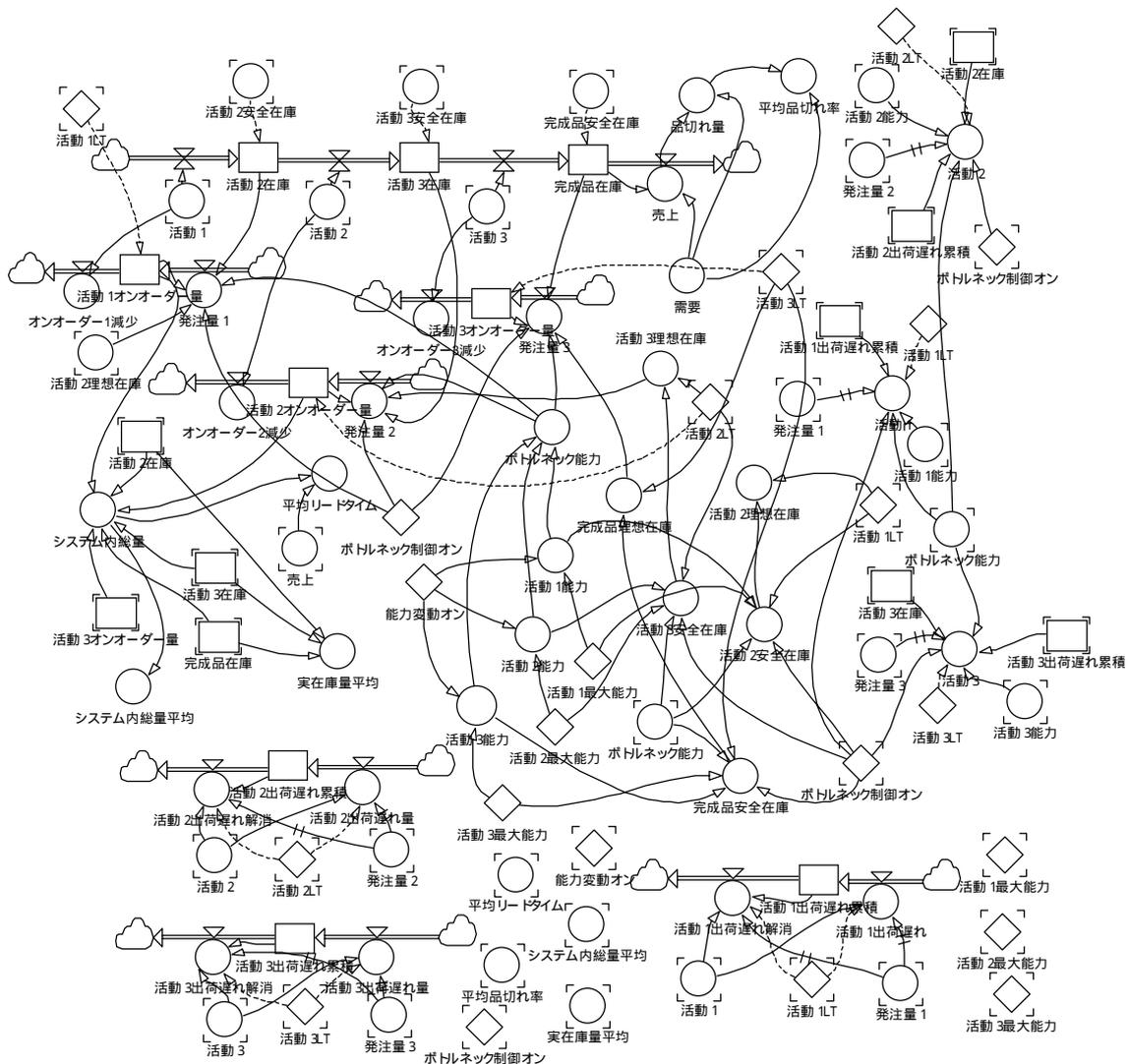
282 頁 例 7 - 7 ボトルネック・モデル 4（独立方式）

モデルの意図

独立方式の場合にボトルネック現象がいかなる影響を与えるかを確認することで、独立方式のチェーン制御における標準性を理解する。モデルは例 7 - 6 プッシュ方式と同じである。

モデルの構造

今度は各々の活動主体が独自に判断して前の主体に発注するような場合のモデルである。



モデル構造とすればプッシュ方式と異なる点は各活動主体においてその適正確保量（「活動1理想在庫」、「活動2理想在庫」、「活動3理想在庫」）と現有量（各主体の活動後の在庫にそれらのオン・オーダー量を加えたもの）の差を前主体に発注する方式になっていることである。

制御の仕方，能力変動，チェーンのパフォーマンス尺度に関してはプッシュ方式の場合と同じである。

シミュレーション演習のポイント

プッシュ方式の場合と同様に，ボトルネック現象に対する独立方式の反応特性を理解することが重要になる。独立方式の安定的なパフォーマンスによってボトルネックにも抵抗力のある特性を理解する。

モデルの意図

前のボトルネック・モデル3，4と同じ状況のモデルであるが，ボトルネック現象に対するプル方式の特性を理解することが目的である。売り上げに応じて活動3が供給し，その活動3の供給した量だけを活動2に発注する。また活動2はそれに応じて動き，活動1に発注する。流出量に応じて前工程に発注する。納入されなかった量は事後的に供給する。

モデルの構造

内容的には各活動主体の発注行動が違うだけでそれ以外は今までのプッシュ方式，独立方式と同じになる。能力変動，各種の制御，リードタイムなどの変更はプッシュ方式のモデル（ボトルネック・モデル3）と同じように行う。

シミュレーション演習のポイント

プル方式は在庫を増やさないという特性で知られている。ボトルネックはボトルネックに適合した量を流すという基本的な対応策があるが，プル方式は基本的にそのような対応策を内包した方式であることを理解する。その意味でプル方式の意義を再確認することが重要である。他方で，それと引き換えに品切れを多く出す可能性を秘めていることも理解しておく必要がある。能力変動などの要素を変化させてプル方式の特性を理解する。

