



簡易マニュアル Powersim Studio 7

POSY Corp. (<http://www.posy.co.jp>) 松本憲洋 (matsu@posy.co.jp)

Contents

1. Powersim 社のモデリング方法論
2. システムダイナミックスとは？
3. Powersim Studio 入門
 3. 1 Studio の概要
 3. 2 操作画面
 3. 3 プロジェクトの概念
4. モデリング・ツールの適用価値と適用領域
 4. 1 なぜ Studio は使われる必要があるか？
 4. 2 Studio の適用領域
5. システムダイナミックスに基づくモデリング
 5. 1 モデルの構造の表現
 5. 2 モデルの構成要素である変数の定義
 5. 3 簡単なモデルの構築と実行 ⇒ [How to build a model?](#)
6. Powersim Studio の特徴
 6. 1 単位付けの効用
 6. 2 組み込まれている時間軸
 6. 3 便利な表示機能
 6. 4 外部データとの接続
 6. 5 戦略シミュレーション
 6. 6 シミュレーターの作成
 6. 7 外部データとの接続 ⇒ [How to connect with Excel?](#)
 6. 8 モデルの再利用を可能にした階層構造
 6. 9 先進的な機能
7. 技術サポート

【利用頻度が高い関数の概説 一覧表】

注1) この資料の最新版は、次の URL からダウンロードできます ⇒ <http://www.posy.co.jp/manual-f.htm>
注2) この基礎的解説資料は Powersim Software AS の資料に基づき、POSY Corp. 松本憲洋が日本のユーザー向けに作成しました。著作に関する全ての権利は両社に所属します。
注3) この資料に関する詳細な技術的内容については、POSY Corp.がおおよそ隔月に開催しているオリエンテーションコースで解説と演習を行っています。 Cf. URL www.posy.co.jp

1. モデリングの方法論

事業計画などのビジネスに適用する Powersim 社のモデリング & シミュレーション・ツールは、システム・ダイナミクス(以後、SD)の論理的枠組みに基づいています。このモデリングでは、システムの構造を体系付けるためにフィードバック理論を適用し、ビジネス・システムの動的挙動を推定するためにコンピュータ・シミュレーションを活用しています。このSDによる方法論は、複数の学問分野におけるモデリング手法が統合されたもので、データの同定、概念的説明、対象世界の描写、ビジネスにおける戦略・戦術・運営問題の分析などに適用されます。

シミュレーション・モデルの価値は、ユーザーがモデルを学習し分析することにより、実ビジネスや実システムにおける問題に対する結論を導くための材料が、実世界に先駆けて得られることです。したがって、シミュレーション・モデルは、実際に事態が発生する前に、マネージャが事態を想像したり、経験したり、管理したりすることをサポートするための道具となります。

マネージャは実システムについて、さらに多くを学ぶために、シミュレーション・モデルを使っているような実地体験をしますから、これはプロトタイプ的な経営決断のための道具とも言えます。例えば、シミュレーション・モデルは、新市場への進出、新商品への取り組み、あるいは、他社の買収や統合によるリスクと利益還元などについて結論を導くために使われます。また、規制撤廃が現在のビジネス・モデルと将来の採算性にどのように影響するかを評価するのもに使われます。すなわち、いかなるコスト削減が、ビジネス・プロセスや組織を変えることによってもたらされるか、ありうるリスクを想定して特定の投資からどのような利益還元が得られるかなどの評価に役立つのです。

シミュレーション・モデルは、動的な方法による実験的な経営ツールであるというだけでなく、分析的なツールでもあります。モデルの構造は、対象となるシステムの構成要素とそれらの関係付け、およびその因果プロセスを含んでいます。このようなシミュレーション・モデルは、組織における意思決定のあらゆるレベルで便利に使われます。マネージャが実システムではなくこのようなシミュレーション・モデルにより計画策定や意思決定における経験を積むことで得られる利点を以下にまとめおきます。

- 構築して操作する時間が短い

- 構築するための費用が少なく、モデル構築の投資回収期間が短い

- 経営の見込みがすばやく認識できるので、直ちにフィードバックをかけられる

- 行動計画の代替案が得られるので、マネージャが直面している問題に対して多数のソリューション候補を準備できる

- 実システムと違って失敗したとしても、リスクは小さい

- 実験を通して学習する意欲が高まる

- 能力あるマネージャの間で連携する意欲が高まる

シミュレーション・モデルの妥当性は、モデルが作られた意図と、設定されている仮説によります。シミュレーションに基づいて導かれた結論の正しさは、モデルが実際の状況やシステムをいかにうまく表しているかにかかっています。モデルと実態とが似通っていればいるほど正しい結論が得られます。ですから、モデラーが立派なシミュレーション・モデルを作るには、情報の有効性、正確さ、

そして信頼性が重要になるのです。企業の中でそのような情報源は、情報システム、書類、各分野の専門家、組織内の意思決定者のメンタルモデル(直感)などに存在しています。

2. システムダイナミクスとは？

システム・ダイナミクス(SD)は、複雑なシステムを分析して理解し、何らかの方法でそれを修正、変更するための手法です。工学分野の制御理論と同根の技術で、あえて短絡的な表現をするなら、SD は制御理論を社会系の対象あるいは問題に適用した技術分野と言えます。したがって、SD とは、システムにおける構造の理論であると同時に、方針設計に対する取り組みの一つでもあります。

SD は次の二つのコンセプトからなっています。

フィードバック理論:システム構造を組織化するための一般的なガイドライン

コンピュータ・シミュレーション:システム構造から生じる挙動を推定するための方法

現在の SD はコンピュータを使って図式的にしかも数学的にモデルを構築します。そのモデルを使った時間軸ベースのコンピュータ・シミュレーションにより、構築したモデルの挙動を観察することができます。その結果、ある時点のモデルの状況が、いかようにして後の時点のモデルの状況に影響を及ぼしているかを理解できます。このコンピュータ・シミュレーションは、構築した図式的なモデルが Studio の中で自動的にコンパイルされて、数学的には非線形・連立・常微分方程式に初期値を与えて解く形で実行されています。

結論として、一般的に SD とは;

時間経過と共に変化する複雑でダイナミックなシステムを学習することに関係している技術

システムの変化が、「なぜ(原因)」、「いかんにして(パターン)」起きるかを発見することに関係している技術

3. Powersim Studio 入門

3.1 Studio の概要と特長

Powersim Studio はシミュレーション・モデルを構築して実行するためのシステム・ダイナミクスに基づく PC 上の統合環境です。システムをモデル化する上で理解し易く他人に説明が容易な図式モデリング言語を使っていますから、できあがったモデルも系統立ち分かり易いのです。これが Studio を使って組み立てられたモデルの大きな利点になっています。

組織や問題に対して図式モデルを描くと構造が定義されたこととなります。次に、個々の構成要素の中身を表現してモデルの挙動を定義する必要があります。それには、Microsoft Excel において採用されている数式に良く似た表現形式を使います。Studio はこの数式を含む全体の操作性がマイクロソフト社の Windows に似せて作られていますから、Windows Office などを操作する感覚で対応すると、直感的に操作できます。

Studio には傑出した特長があります。その一つは、外部のデータベースとの接続機能です。Studio はマイクロソフトの Excel と SAP 社の NetWeaver BI との間で直接的に情報交換するこ

とが可能です。これにより、実世界のビジネス・データをモデルに呼び込み、そしてシミュレーション結果である将来のシナリオなどを実世界へ持ち出すことができます。

次の特徴は、組み込まれているリスク評価の機能です。この機能を使って最終的に導かれたモデルに対して感度分析を実施することで、対象としたシステムの不確実性を、モデリングとの一貫性を保ちながら容易に評価できます。

そのほかの特長を以下にまとめます。

- ▼ 階層構造と配列によりもたらされる高度なモデリング機能
 - ユーザーに対して、操作性と説得力に優れた I/O インターフェース
 - 図式的なモデリング言語と Excel で用いている数式に良く似た数学的な言語
 - 計測値の単位により内部で自動的に論理チェックをサポートする機能
 - グラフやテーブルのような強力なデータ表現法と、データを容易にその表現形式へ連結する機能
- など

3.2 操作画面

操作画面として2種類があります。それぞれの名前は、「Normal」と「Advanced」です。両者の違いは、左側のプロジェクト・ウィンドウと名付けられているシミュレーション・モデルの構造を表現しているウィンドウの有無です。

“Advanced トップ画面”

“Normal トップ画面”

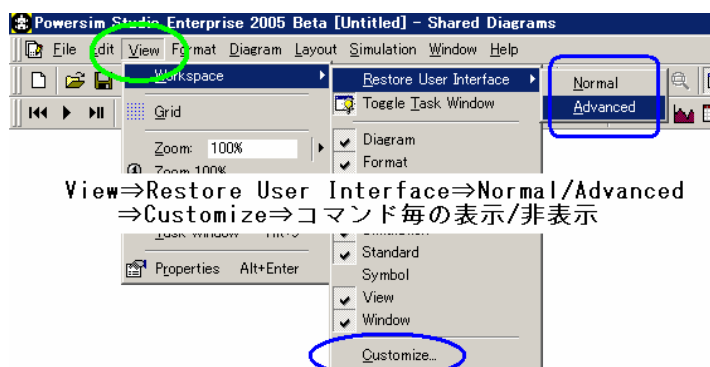


高度な操作画面

初心者向け操作画面

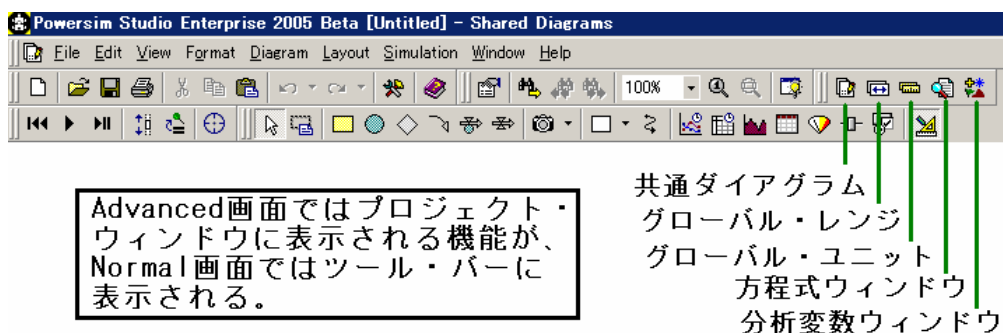
このプロジェクト・ウィンドウには、一つのファイルに保管される複数のモデルの階層構造や外部データベースとの関係を定義する機能が含まれていますから、Advanced を使うと大きくて複雑なモデルを構築できます。それに反して Normal ではただ一つのモデルしか構築できませんから、初心者向けの学習用のトップ画面といえます。

トップ画面の切り替えは、下図に示すように、メニューバーの View を開き、Workspace → Restore User Interface → Normal/Advanced を選択します。また、両者のトップ画面のメニューバーとツールバー上のアイコンの配置をカスタマイズする機能もついています。



以上で説明した方法のほかに、画面の右側に表示されるタスク・ウィンドウの下方にある“Change workspace”からも画面の切り替えができます。

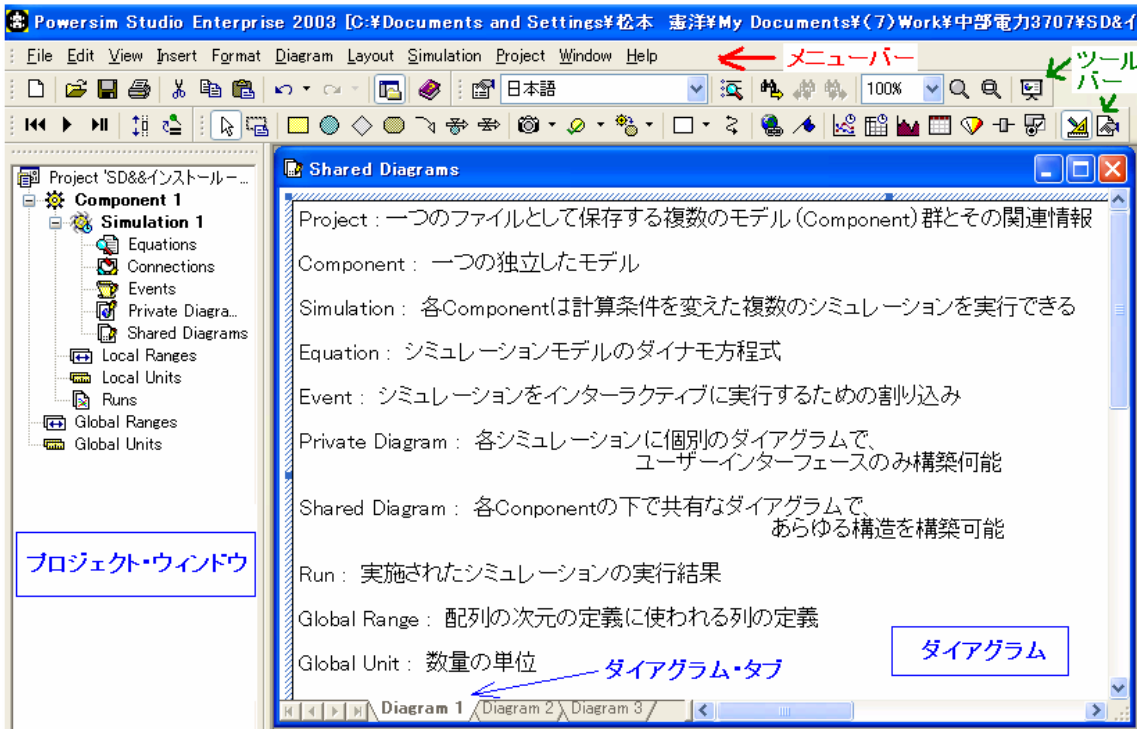
なお、Normal トップ画面では、プロジェクト・ウィンドウが開かないので、ツールバー上に最小限必要な機能が下記に示すようにアイコン化して配置されています。



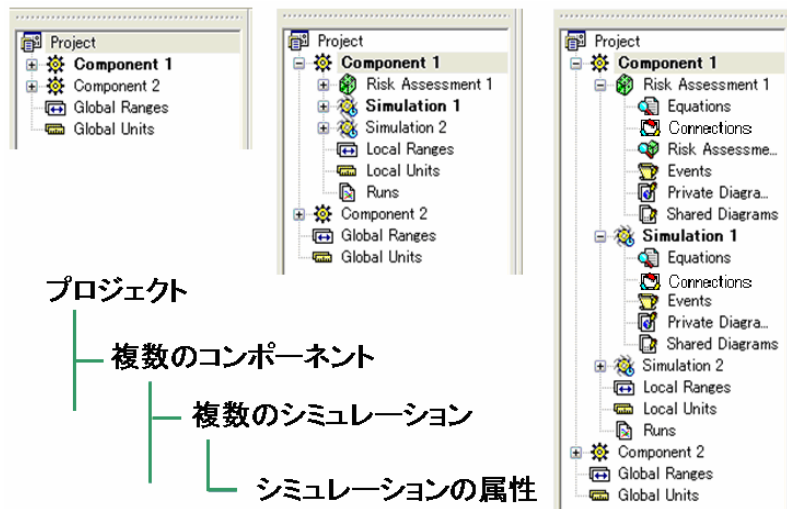
3.3 プロジェクトの概念

問題解決に取り組む一般的なプロジェクトを想定してみてください。大きなプロジェクトは複数のコンポーネントに分解されます。それぞれのコンポーネントは階層構造を形作って相互に影響しあう場合もありますが、コンポーネントそれぞれにリーダーがいて、全体の問題解決に向けて独立した形態で活動します。各コンポーネントでの活動が終結に近づくと、外部条件の入力方法や結果の表現方法に特徴がある複数のコミュニケーション方法が準備され、相手によって異なる方法を用いることで説明・説得効果を高める工夫をします。



Studio における問題解決にも同じような仕組みが準備されています。Studio の Advanced 画面を立ち上げると、上端のメニューバーとツールバーの下に、3つのウィンドウが開きます。左側がプロジェクト・ウィンドウで、この中に表示されている木構造全体が一つのプロジェクト(または、シミュレーション・プロジェクト)と呼ばれていて、このウィンドウに表示されているのはプロジェクトの構成です。一方、真中はダイアグラムで、ここではコンポーネントごとに(場合によってはサブモデルごとに)モデルの構造と個々の要素の数式が定義されます。ダイアグラムは何枚でも持つことができます。これは Excel の Sheet の取り扱いに似た操作により下方のタブで切り替えながら使います。



コンポーネントとは単独にシミュレーションできる独立したモデルのことです。ファイルとして保管する場合には、このプロジェクト全体でプロジェクトの名前の後ろに拡張子“sip”を付けて保管します。プロジェクト・ウィンドウの一例と簡略化した構造とを以下に示します。



最後に、右側のウィンドウがタスク・ウィンドウで、この中では、最近使ったモデル、学習と演習の教材、サンプル・モデル、ヘルプ機能などを容易に使うことができます。

プロジェクト・ウィンドウとタスク・ウィンドウは、ツールバーにあるアイコン  と  で、表示と非表示の切り替えができます。

4. モデリング・ツールの適用価値と適用領域

4.1 なぜ SD ツール“Studio”は使われる必要があるか？

あらゆるビジネスにおいて、運営上の詳細部におけるオペレーションから全社戦略の策定に至るまで、各レベルの意思決定のために“モデル”は毎日使われています。例えば、我々が問題に直面したとき、望ましい結論に到達できるように因果を踏まえて仮説を置きながら解決への道筋をたどって行きますが、その過程で必ず自分自身のメンタルモデルを何らかの形で使っています。

しかし、メンタルモデルに基づいて意思決定にたどり着く知的行動をとろうにも、我々のメンタルモデルはあまりに複雑ですから、我々自身の知的レベルをはるかに超えてしまっていて論理不明確に成らざるを得ないのです。その結果、途中で論理的な追求をあきらめ、勘と度胸の決断に甘んじることも多いのです。このことが、コンピュータ上でメンタルモデルに代わる人工モデルを描き出すために、Powersim ソフトウェアのモデリングとシミュレーション・ツールである Studio がなぜ使われているのかということの答えを示しています。すなわち Studio によれば、我々が解決への道筋で置いた仮説のもたらす短期・長期の影響を踏まえて、明確な結論がたちまちのうちに計算され表示されるのです。

このように、Studio を使ってシミュレーション・モデルをビジネスの場に適用するならば、仮想経営が可能になりますから実ビジネスにおける実践の前に、マネージャは PC 上で戦略を開発し仮想的に試してみることができます。その結果、戦略の選択に基づくリスク回避と利益還元に関する調査に要する時間と労力が大幅に削減されます。そればかりか、ビジネス構造、戦略、短・長期の業績の間の因果関係を理解したり改善したり新たに関係付けたりする能力も高められますので、実ビジネスにおいて成功する確率が上昇します。また、コンピュータ・シミュレーションのインタラクティブな効果により、さまざまなビジネス条件の下でマネージャが動的な状況を体験する環境は刺激的ですから、ケーススタディや書類上の静的なシミュレーションに比べて学習効果ははるかに高くなるのです。

4.2 Studio の適用領域

Powersim 社のモデリングとシミュレーション・ツールは、欧米で広範囲な産業領域で使用され、期待にたがわず成功をもたらしています。具体的には、世界的企業であるマイクロソフト、フォード自動車、プリティッシュ石油、マクドナルド、ヒューレットパッカードなどで、戦略立案、資源の運用管理、危機対策と管理、プロセス・リエンジニアリングなど広範囲な問題に対して適用されて、望ましい解決案を導いてきました。

日本では 1970 年ごろの SD ブームを除くと、SD が実社会に適用された事例は少ないようですが、現在のグローバルな社会・経済の下で日本だけが例外ということはありません。今後は PC 上の仮想空間において目指す戦略経営を予行し学習を重ねた上で、その結果を実空間での実際の経営に活かすというプロセスが必要になるでしょう。いわば、仮想経営を組み合わせた仮説検証型経営です。

以降では各領域で適用されてきた Powersim 社のモデリングとシミュレーションの効果について概要を述べます。

企業経営への応用 シミュレーションは、その企業の運営、競争相手そして市場を踏まえたモデルを構築することにより企業経営へ適用することができます。効果的なシミュレーションを利用することで、従業員は方針分析、競争戦略、シナリオ分析などが、リスクの脅威がない PC 上で可能になります。そして彼らが毎日行う意思決定結果に関する識見を得ることもできます。

工業への応用 生産加工や工場設備において見出される生産、輸送、流通のシミュレーション・システムによって、Powersim ソフトウェア製品を工業へ適用することができます。シミュレーションのプロセスに現場のデータを連結した統合型モデルを開発することにより、いかなる内外の力が生産性と採算性に影響するかを観察し分析することができます。

環境問題への応用 Powersim ソフトウェア製品は、生態系や環境問題への人工的な介入の影響をシミュレーションすることで、環境問題に適用することができます。それにより代替シナリオを策定でき、それらの時間経過における影響について見通すことが可能になります。

意思決定への応用 Powersim ソフトウェアによる意思決定シミュレーションを適用すれば、従来の戦略計画過程に対して「力強さ」、「柔軟さ」、「速さ」を付加できます。Powersim のシミュレーションは、ビジネス自身とビジネス戦略との総合的なコンピュータ・モデルに基づき、組織の KPI における将来の変化を推定します。

経営教育への応用 Powersim の経営訓練シミュレーションは、経営教育を実施するための強力なツールです。受講者に擬似化されたビジネスの仮想世界の中で、経営フライト・シミュレーターを実地体験させることにより、伝統的なワークショップとイー・ラーニング・プログラムとを補完できます。

価値伝達のためのシミュレーション 価値伝達のシミュレーションは、新しい企業の開始や冒険的事業についてビジネスの行く末を評価したり伝えたりするための直感的な方法です。このようなシミュレーションにより、企業が将来出くわす可能性があるビジネス上の成功や潜在的なリスクに関する重要な要因についての洞察することができます。また、価値伝達のためのシミュレーションは、マネージャが全従業員に、新しい戦略方針の理論的な根拠を伝えるためにも役立ちます。

5. システムダイナミクスに基づくモデリング

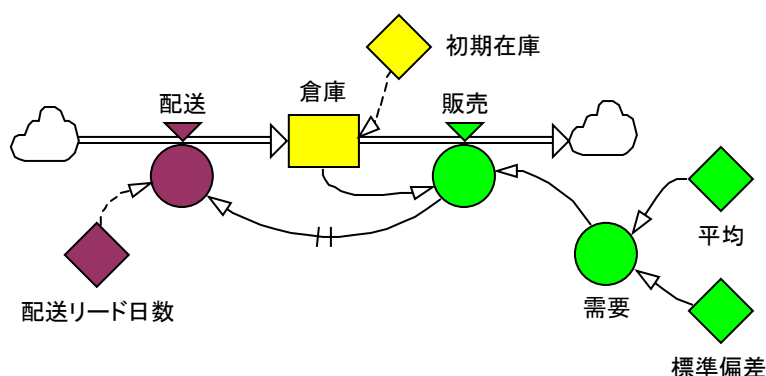
5.1 モデルの構造の表現

行と列を使って作る伝統的なスプレッドシート上のモデルと違って、Studio では図式的なモデリング言語を使ってモデルを作ります。また、Studio には時間軸ベースに計算する仕組みが組み込まれていますから、時間経過のシミュレーションを自動的に実施できます。したがって、スプレッドシートでは別に準備しなければならない時間設定を Studio でモデルを作る場合には自分で組み込む必要がありません。

各種の変数シンボルと接続シンボルを使うことで、直感的で説明的なモデルを作ることができます。そのためモデルの論理が理解しやすく、そしてまた、他人に説明するのも容易です。ダイアグラムは Excel シートと同様に一枚ずつが体系付けられ、それぞれのシートがモデルの一部を表しています。また、お好みによりシートはシミュレーションのインターフェース作りにも使われます。

モデルの中の時間的な考え方に対処するために、何種類かの型式のシンボルを使ってモデルを組み立てる必要があります。時間経過に伴う値を蓄積する変数はレベル(■)と呼ばれます。レベルはその時点のシステムの状態を表しています。例えば、顧客、在庫、作業量、預金口座などがそれに相当します。レベルはシミュレーションの開始時に初期化されますが、シミュレーションの実行中には、フロー(⇄)だけがレベルに入出力して影響を与えることができ、レベルの値はフローによってのみ増えたり減ったりします。フローはシステムの中の活動を表していて、例えば、潜在顧客から実顧客への移行の流れ、在庫に対する商品の移出入、そして銀行口座に対する入出金などがこれに相当します。

Studio にはシミュレーションのステップ(刻み時間)ごとに毎回計算されるタイプの補助変数(●)も含まれています。補助変数はフローを制御したり、中間の計算を実施したりします。変数間の依存関係は情報リンク(→)を使用することで表示されます。そのほか Studio は定数変数(◇)も含んでいて、この変数はシミュレーションの間中、値が変わりません。定数は、モデルの各レベルを初期化する値、変数間の関係の強さを定義する値、あるいはユーザーがモデルに入力オブジェクトを介して入力する値のために使われます。



上図は、モデル構造がどのように見えるかを示した例です。フローの左右の雲の絵は無制限であることを示していて、この外はモデルの範囲外であることを意味します。実線は情報のリンク線で、点線は変数の初期化に使われたリンク線であることを示しています。

5.2 モデルの構成要素である変数の定義

図式モデルではシステム内の変数の構造を表現しました。モデルの定義を完成するために、構成要素である変数に数学的な定義を加える必要があります。Studio の数学的な定義言語は Excel で使われている数式に大変よく似ています。ただ、Excel では変数の位置も名前もセルの位置により固定されていますが、Studio ではセル名を参照するのではなく、ユーザーが分かりやすく定義した変数名を使います。例として、一人の潜在顧客が実顧客に出会う確率の計算式を以下に示します。

$$\text{方程式} = (\text{顧客} / (\text{顧客} + \text{潜在顧客})) * \text{潜在顧客}$$

Studio の数式の定義言語には、計算で使用される広い範囲の関数が含まれています。これらは、



統計、数学、財務、三角関数など、200 種類以上に及びます。また、ユーザー特有の関数も特殊な関数である VBFunction を用いることで、VBScript により書き込むことができます。

5.3 簡単なモデルの構築と実行

簡単なモデルの作り方とシミュレーションのやり方について説明します。モデリングへの取り組み方の基本は、モデリングの構想の中の重要な部分から先ず手がけてみることです。そこを手がかりにして、周りにモデルを拡大していくやり方で一つのモデルを構築します。そのようにして構築したモデル(Studio の言葉では、コンポーネント)を階層構造により結びつけることで、シミュレーション・プロジェクト(全体モデル)を整えていきます。簡単であっても何らかのモデルを作り始めると、Studio のオンライン・ヘルプからの情報や、インタラクティブなコミュニケーションによる知識の獲得により、モデリング技術の修得も進みます。ここではビギナー向けの単純で簡単なモデリング技術のみの修得をめざしてみましよう。したがってここでは、モデルを構築するために必要な最低限な技術だけについて説明します。その結果、初歩的な SD と Studio に関する修得により、既に出来上がっている多くのモデルに条件を与えて実行することができるようになります。例えば、POSY 社の下記の HP から自由にダウンロードできる既存のモデルを沢山に試みていただき、次の技術レベルを目指してください。

参照モデルのダウンロード: <http://www.posy.co.jp/model-distribute2118-f.htm>

なお、POSY 社が開催するオリエンテーション・コースに参加すれば、SD とモデリング & シミュレーションの初歩技術を効率的に修得することができます。

(1) Studio のインストール

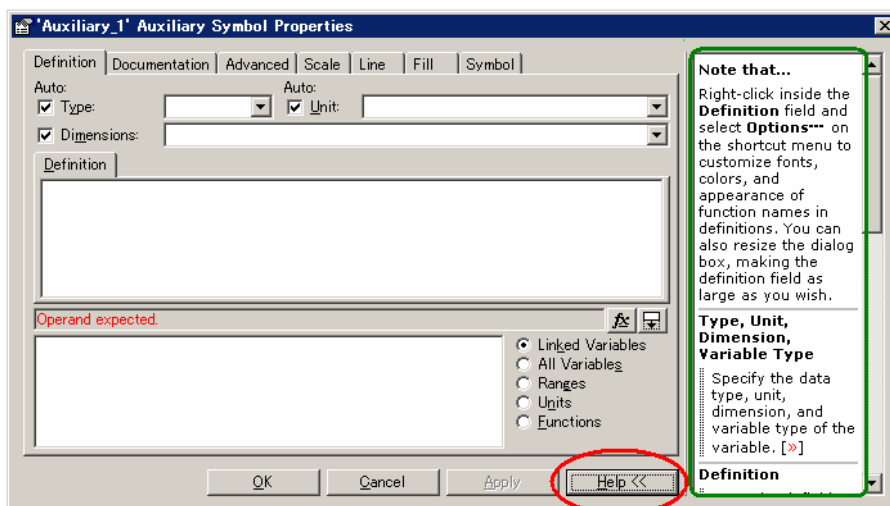
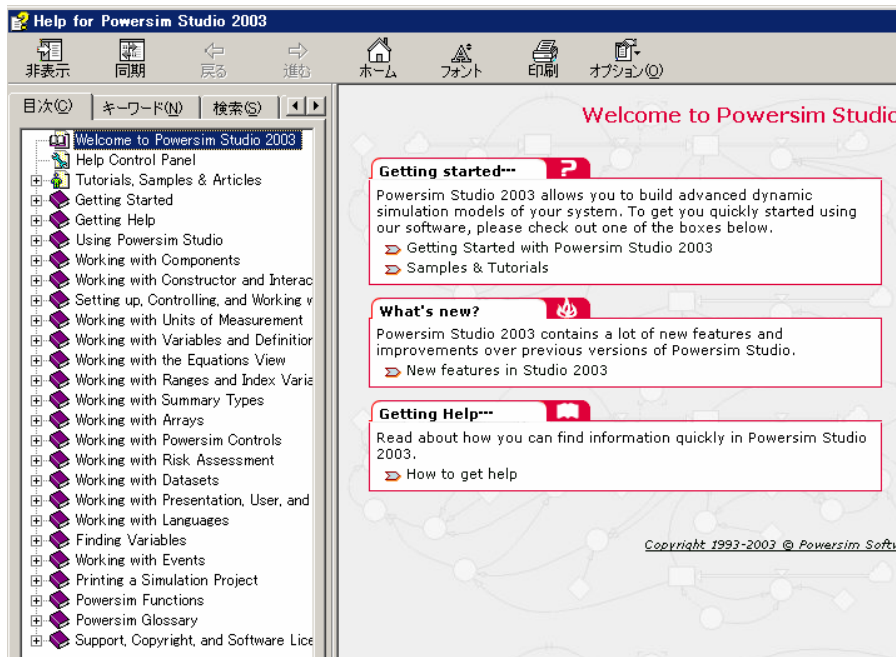
Studio をインストールするためにはバージョンにかかわらずプロダクト・キーが必要です。Powersim 社の HP からフリーの Studio Express をダウンロードすると、2ヶ月間有効なプロダクト・キーがメールで送られてきます。有効期間が無効になったら何度でも入手して下さい。同じ PC 上で Studio Express を繰り返し使用することができます。また、ここで開発したモデルは保管できて、後日に商品版上でも活用できます。

(2) マニュアル

Studio のメニューバーの右端にある HELP あるいはタスク・ウィンドウの中ごろの HELP をクリックし、プルダウンメニューから一番上にある Contents を選択します。

“目次”、“キーワード”、“検索”のいずれかから対象の単語を入力して条件にヒットしたページを表示して参照して下さい。

プロパティ画面などの下端に HELP ボタンがついています。このボタンを ON すると、そのプロパティ画面に出ている単語の説明が右側に表示されます。この画面は、大きさを変えたり、プロパティ画面から切り離して配置したり、便利に利用できます。

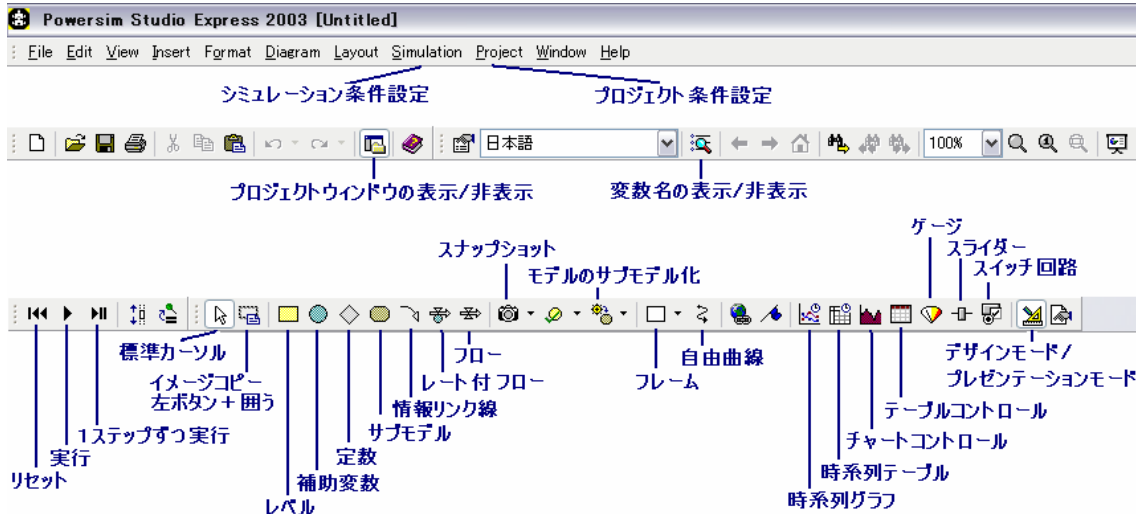


Studio 全体のマニュアルは、以下の URL に保管されていますのでご利用ください。

<http://www.powersim.com/download/default.asp>

(3)メニューバーとツールバー上の主要なアイコン

最低限必要な機能にリンクされているアイコンについてのみ、以下の画面で簡単な使い方を説明します。ただし、マイクロソフトのソフトウェアと同じ機能のアイコンについては説明を省略します。

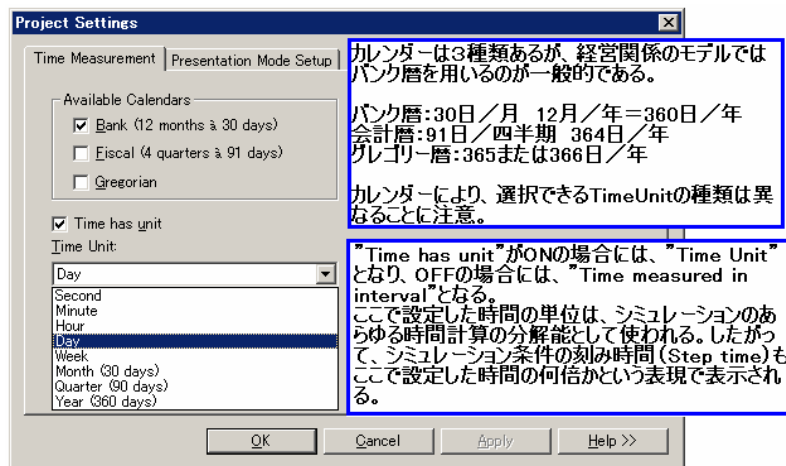


(4)シミュレーション条件の設定1:Project Setting

メニューバーの“プロジェクト”のプルダウンメニューから、最下段の“Project Setting”をダブルクリックして、下図のウィンドウを開きます。このウィンドウでモデルの時間軸の条件を設定します。

Studioでは変数の単位を重視して、モデルの中で単位計算することで論理チェックを行っています。時間軸についても同じで、例えば、利息の計算において、年利で $r\%$ / 年と設定すれば、計算の刻みが変わっても自動的に変換して計算するようになっています。

しかし、望ましくはありませんが、操作を簡便化するためにモデルの論理的な確実さを犠牲にして、従来のツールのように時間単位を定義しないでシミュレーションを実行することも可能です。その場合には、下図の“Time has unit”のチェックをはずします。ただ、この場合もフローレートの中だけで時間単位を1と考えて、シミュレーション条件の設定や時系列グラフ表示などでは、“Time measured in interval”で選択した時間単位が、時間の分解能(時間の原子単位)として取り扱われます。



(5)シミュレーション条件の設定2:Simulation Setting

メニューバーの“シミュレーション”のプルダウンメニューから、最下段の“Simulation Setting”をダ

ブルクリックして、下図のウィンドウを開きます。このウィンドウでシミュレーション条件を設定します。この画面の中の、Time、Timestep、Starttime、Stoptime は時点を示すためにあらかじめ定義されている変数ですから、ユーザーは他の意味で使うことができません。時間については以下の略号が定められています。

yr:年 qtr:四半期 mo:月 wk:週 da:日 hr:時 min:分 s:秒

なお、1日(24時間)以内の時刻は、日付の表示の後ろに半角文字の空白を開けて次の表現で記載します。なお、この前の日付の表示を略すこともできます。その場合には、西暦0年1月1日(0000/01/01)が省略されているとみなされます。

hh:mm:ss.ss hh:時間 mm:分 ss.ss:秒



(6) 単位の設定

前の(3)と(4)で時間軸に関連した設定について説明をしました。次に、時間軸以外の変数の単位設定について要点を説明します。

Studioのプロジェクト・ウィンドウには、プロジェクト全体について定義するGlobal Unitとコンポーネントごとに設定するLocal Unitがありますが、一般には、Global Unitだけで済みます。

Global Unitをダブルクリックして画面を開きますと、既に定義されている単位が表示されます。

空白の部分で右クリックすると、Add UnitとAdd Empty Unitが表示されます。

Add Unitを選択するとダイアログ・ボックスが開き、Enter New Name、Select Built-in Unit、

Select Built-in Currency Unitを選択するためのラジオ・ボタンが表示されます。貨幣の単位と標準単位については2番目か3番目を選択し、カタログされていない単位については、最初のEnter New Nameを選択して名前を入力する必要があります。

ここでは次の(6)の簡単なモデルに必要な商品の個数の単位“ko”を設定しておきます。単位はダブルバイトで表現しても構いません。しかし、変数のプロパティ画面の中で方程式の文字はシング

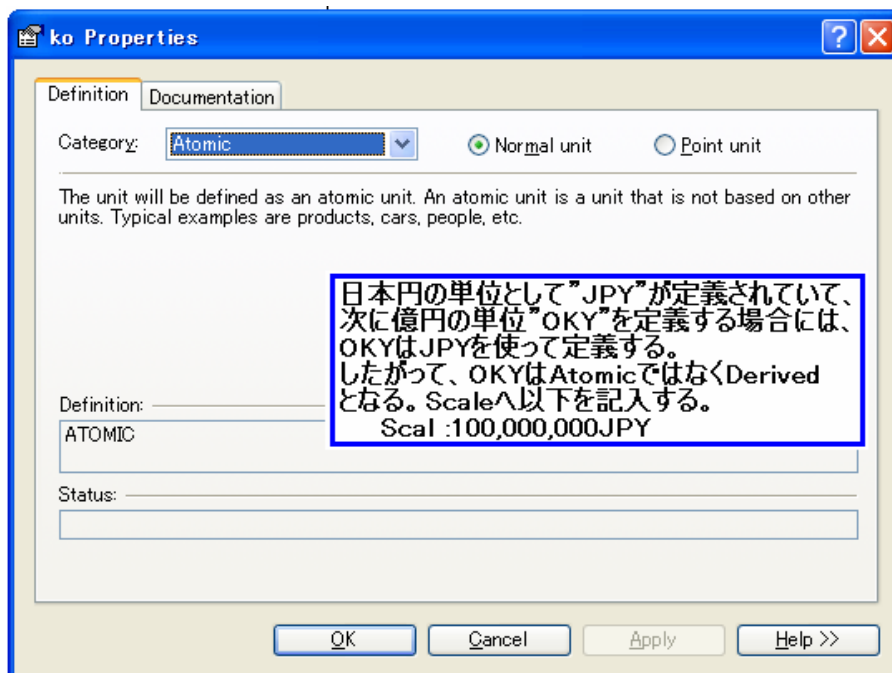
ルバイトに限られていますから、単位がダブルバイトだと一つの画面でのシングル/ダブルの切り替えが頻繁に起こり、間違いが発生する原因になります。したがって、単位はシングルバイトで定義することをお勧めします。

新しい単位を定義する過程で次の2つの特殊な内容があります。

①商品の個数の単位“ko”は、特定な点を計測する単位ではなく、相対的な大きさを測る単位なので、point unit ではなく、normal unit を選択する。

point unit と normal unit とは、例えば、温度の場合、地球上の1気圧では、0 deg.C で凍結し、100deg.C で沸騰するから、point unit としてのセ氏の計測温度の単位は“@C”で、相対的な温度差など normal unit としての温度単位は、“C”で表す。

②個数“ko”は最小分解単位であるから、Atomic(原始単位)を選択する。



以上の(3)～(5)で説明しましたように、モデルの各要素にそれぞれの単位を定義すると、モデルの中で単位計算が自動的に行われ、それによりモデルの論理チェックがなされることとなります。しかし、単位を1と考えて、個別の単位を付けないでモデルを作ることもできます。そのような場合には、モデルの論理的な確実性をモデルの作成者自身が何らかの方法で十分に確認しておく必要があります。

(7) 基本要素を配置してモデルを構築

簡単な販売モデルを描いてみましょう。

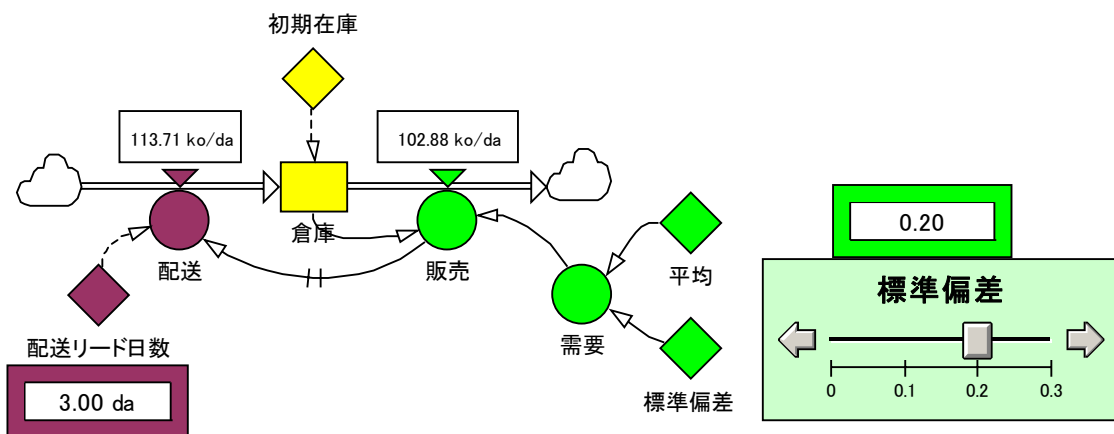
①Studio を白紙の状態を開く

②モデルのストーリー :

平均すると毎日100個の商品が売れている店です。販売のばらつきがあり、平均の20%程度の標準偏差で表せます。即ち、需要を平均100個、標準偏差20個の正規分布乱数で表現しま

す。店主は毎日売れた量だけその晩に Fax で注文します。卸業者は注文を受けてから3日目に配達します。店主は初期在庫をおよそいくら持って、店の営業を始めればよろしいでしょうか？

- ③**店の倉庫** : ツールバーのレベルにカーソルを合わせて左クリックし、白紙のダイアグラムの中に持ち込み、適当な場所で左クリックして画面上に配置します。
- ④**配送** : ツールバーのレート付きのフローを左クリックして、先に置いた倉庫の左側で左ボタンを押さえたまま、カーソルの先に付いているクロスがレベルの中に入るまでドラッグし、左ボタンを離します。
- ⑤**販売** : 配送と似た操作ですが、出発点ではクロスが倉庫の中にあり、右の端ではシンボルからカーソルを離すためにダブルクリックします。
- ⑥**モデル構造の完成** : そのほかの補助変数と定数のシンボルを配置して、シンボル間を情報リンク線で結びます。



- ⑦**外生変数の設定** : 数式の設定のためには各変数のプロパティ画面を開く必要があります。プロパティ画面を開くには、常にその対象をカーソルでダブルクリックします。
配送リードタイムは3日ですから、3<<da>>と definition 枠に入力します。
ここで<< >>で日の記号 da を囲んでいるのは、単位を表す表現法の約束事です。
 - ⑧**レベルの初期値の設定** : レベルについては初期値を入力します。倉庫のプロパティ画面で definition 枠の下にこのシンボルにつながっている変数として、初期在庫の名前が表示されています。この値が倉庫の初期値ですから、初期在庫の文字をダブルクリックしてください。初期在庫という変数名が definition 枠内に表示されます。
 - ⑨**補助変数の設定** : 配送のプロパティ画面ではパイプライン遅れ関数を定義します。

$$\text{delayapl}(\text{被遅れ変数名}, \text{遅れ日数})$$
 需要のプロパティ画面では正規分布乱数を定義します。

$$\text{normal}(\text{平均値}, \text{標準偏差})$$
- ⑦⑧⑨で定義した式をまとめて以下に示します。

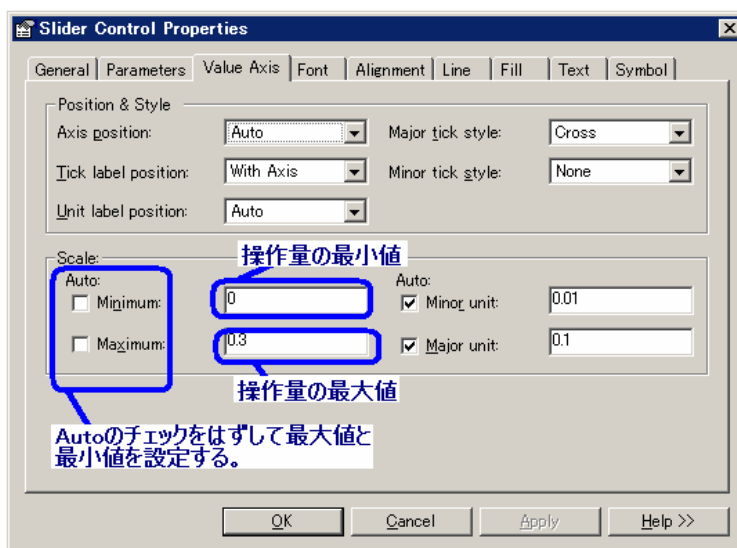
Name	Dimensions	Unit	Definition
倉庫		ko	初期在庫
販売.out			販売
配送.in			配送
初期在庫		ko	300<<ko>>
平均		ko/da	100<<ko/da>>
標準偏差			0.2
販売		ko/da	MIN(倉庫/TIMESTEP,需要)
配送		ko/da	DELAYPPL(販売,配送リード日数)
配送リード日数		da	3<<da>>
需要		ko/da	NORMAL(平均,平均*標準偏差)

⑩入力インターフェースの設定 :ここではテーブルコントロール  とスライダー  の設定を行

います。テーブルコントロールは、シミュレーション実行中は、その時点の値を表示します。テーブルコントロールが定数変数に対応している場合には、シミュレーション終了後にリセットボタンを押すと、表示した変数の値を一時的に変更する入力装置になります。

このモデルでは、配送リード日数と需要の標準偏差に対応したテーブルコントロールを設定します。やり方は、補助変数を配置する場合と同じですが、枠の広さは対角線向きにドラッグして決めることができます。変数の値を表示する枠数の増減は、配置したテーブルコントロールの枠を右クリックするとその操作画面が表示されます。テーブルコントロールに変数を対応させるには、変数のシンボルをクリックしてテーブルコントロールの枠内に引き込みます。テーブルコントロールの特長は、スライダーと違って数値を厳密に設定できることです。

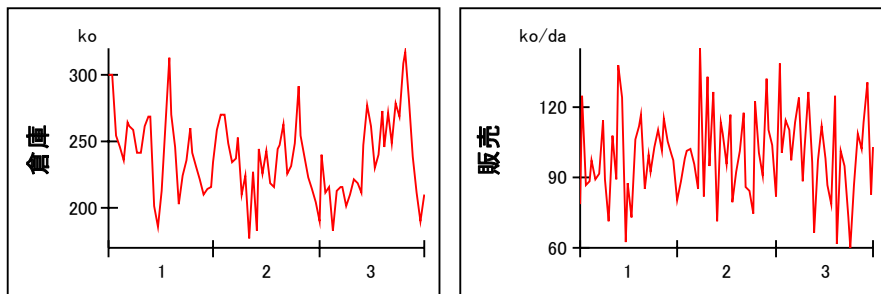
一方のスライダーは、定数変数に対応している場合には、純粋な入力装置になります。テーブルコントロールと同じように、ダイアグラムの中にスライダーを配置して、定数変数をクリックしてその枠内に引き込みます。スライダーは入力装置として操作性がよいのですが、設定全範囲については厳密に数値を設定することができません。そのために数字を厳密に設定するためには、標準偏差に関して設定しているように、テーブルコントロールとの組み合わせが有効です。



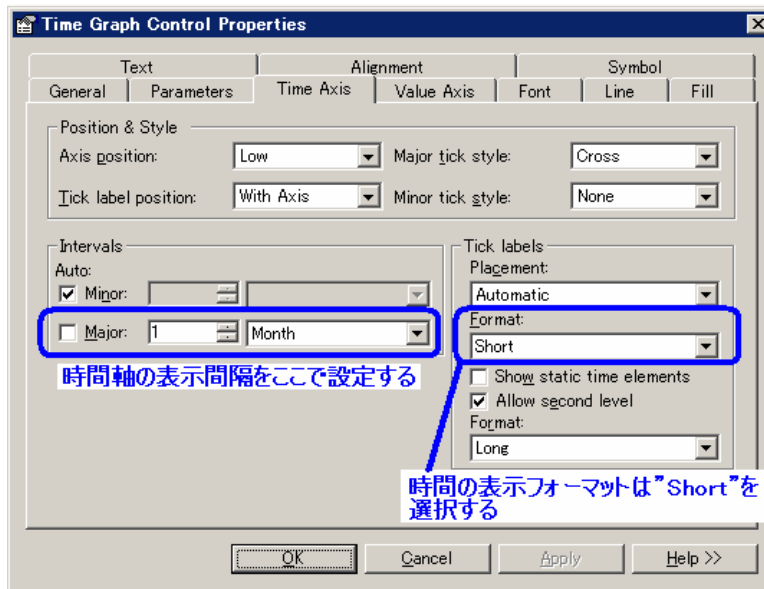
変数やこれらのコンロールに共通していることですが、そのプロパティを設定するにはそのオブジェクトをダブルクリックします。表示されるプロパティ画面には、ほとんど共通な表記方法が採用されています。

(7)シミュレーションの実行と結果の表示

- ⑪時系列グラフ : ツールバーから時系列グラフを左クリックで選択し、モデルの下の適当な場所で左ボタンを押しグラフの対角線方向に引っ張り、適当な大きさになったところで、左ボタンを離します。そのグラフの中に倉庫を左ボタンで押さえたままドラッグしてグラフの中でボタンを離します。ドラッグ中にモデルの絵が変形しますが、何の問題も起こりませんから作業を続けてください。同じことを販売についても行います。



時系列グラフのプロパティも、“問題とする対象をダブルクリックする”という原則に則り、時系列グラフをダブルクリックすると表示することができます。

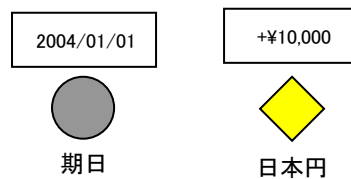


- ⑫シミュレーション条件の設定 : simulation setting を開き、例えば、timestep=1da、Starttime=2007/1/1、Stoptime を 2008/1/1 に設定します。
- ⑬シミュレーションの実行 : リセットボタンを押した後に、実行ボタンを押します。
- ⑭最小初期値を求める : 初期在庫の値を変更して、品切れが起こらない最少の初期条件を探索して求めます。(販売—需要)<0 が品切れを意味します。

6. Powersim Studio の特徴

6.1 単位付けの効用

一般に、何を計測したのか知らなくて単に数値だけ知らされても無意味です。ですから Studio では、どんな数値も数字と単位を組み合わせで取り扱います。計測値に単位を添付すると多くの利点があります。Studio には数値の一貫性を保ちながら照合する機能が組み込まれていますので、矛盾した単位の使い方は自動的に防ぐ仕組みになっています。例えば、人間と製品の計測単位には互換性はないので、(人間+製品)のような矛盾は許されません。また、日付や通貨のように出力形式に約束事がある変数については、以下に示す特別の書式が準備されています。



Studio では変数の定義から変数の単位を推定しています。例えば、二つの変化した距離から定義された変数である速度は、距離はキロメートルで計測され、所要時間が時間で計測されていますと、速度は自動的に“km/hr”を単位として持つこととなります。

したがって、単位を入力するのは、定数とレベルの初期値だけで、その他の変数については内部で自動的に単位が計算されます。しかし、数値の桁数の問題などで、単位を変更したい場合もあります。例えば、走行距離をメートルで計算して表示していたが累積値が大きくなって、キロメートルで表示した方が分かりやすい場合などです。こんな場合には、希望する単位を指定して使うこともできます。ただし、その単位がもともとの単位とモデル内で整合性が取られている必要があります。単位の整合性とは、単位の間で換算式が定義されているということです。

単位を定義する Add Unit のダイアログ・ボックスにおいて Definition のページを開き、基本単位 (Base Unit) の分類 (Category) は原子単位 (Atomic) と設定しますが、関連付ける単位の場合には、分類を誘導単位 (Derived) と設定して、基本単位を選択し、それとの間の換算係数を入力します。

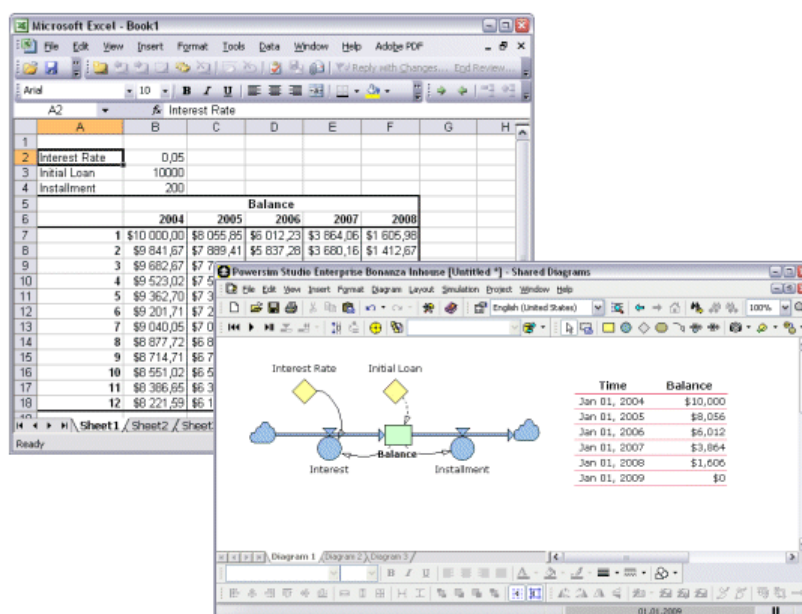
計測値の単位はシミュレーション・プロジェクト全体で定義する場合と、コンポーネントごとに定義する場合とがありますが、前者の Global Units として定義するのが一般的です。

6.2 組み込まれている時間軸

Studio は組み込み時間軸をベースに計算する仕組みを持っているから実行のボタンをクリックするだけで時間経過に沿ったシミュレーションを実行することができます。スプレッドシートを使った仕事を常時やっているモデラーにとっては、Studio のように時間軸が組み込まれているモデリング・ツールは簡単だと感じられることでしょう。

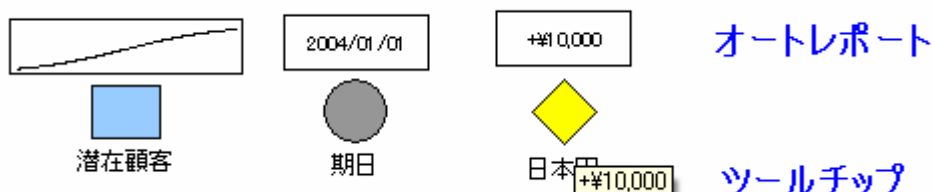
Studio では、モデルの構造を即座に見ることができるのに比べて、スプレッドシートではそれらは隠れていて見えません。また、上述のように時間経過に沿った計算はスプレッドシート・モデルで

は別に時間軸の定義を組み込まねばなりません、Studio ではその必要がありません。以下の図は、スプレッドシート・モデルと Studio におけるシミュレーション・モデルとを対比して、後者の利点を視覚的に説明しています。



6.3 便利な表示機能

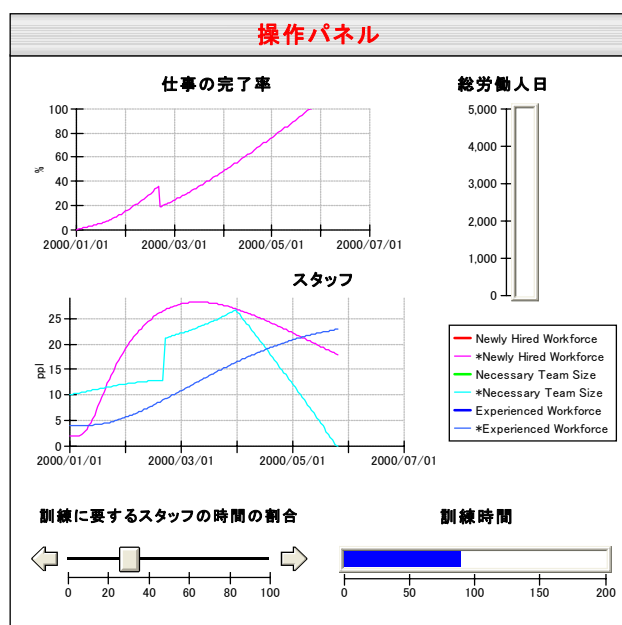
シミュレーション結果は Studio の中で、さまざまな方法で表示されます。モデル上ではカーソルを変数のシンボルに近づけると、ツール・チップにより、その変数の数値が短時間だけカーソルのそばに表示されます。さらに、変数の値を定期的に調べるためには、対象の変数を右クリックしてプルダウン・メニューを表示し、ショー・オート・レポート機能を選択して使います。



オート・レポート機能には、変数の履歴を示すグラフ表示と、シミュレーション時点ごとの数値を示す機能とがあります。I/O のインターフェースを兼ねて、もっと永久的に表示するには、Studio の中に組み込まれている表示オブジェクトを使います。これらは時系列グラフ、テーブル、スライダー、チャートなどですが、それらのいくつかが組み込まれている例を以下の説明図に示します。

もし、表示オブジェクトにリンクされている変数がユーザーからの入力を受け付ける変数なら、その表示オブジェクトはデータ入力オブジェクトとして使われます。下の説明図で、下端に並んでいる二つのオブジェクトは共にスライダー・オブジェクトです。左側のスライダーは、ユーザーの入力

を受け付けています。というのは、このスライダーにリンクされている変数が定数だからです。このスライダーのハンドルの位置を動かすと、その位置に相当する数値が入力データに反映させられます。一方、右側のスライダーはユーザーの入力を受け付けていません。というのは、これにリンクされている変数が補助変数だからです。スライダーは入力オブジェクトとしては使われないインディケータ・バーに自動的に変わっています。



[<元へ戻る>](#)

6.4 外部データとの接続

Studio は外部のデータと接続するためにデータ入出力用のデータセットを持っています。この機能により、モデルは実世界のデータを読み込んでモデルの中で利用することができます。また、シミュレーション結果を更に解析、保管あるいは表示するためにモデルの外に転送することも実に容易です。モデルが実挙動に一致するように、そのパラメータをチューニングするには時系列データを自由に入出力できる機能が必要です。また、将来のシミュレーションを実行する場合には、その時点に相当した仮説をこの機能により、シミュレーション条件として持ち込むことができます。その際、シミュレーション・モデルそのものは一切変えないで、いろいろなシナリオに沿って外生変数などを外部から取り替えながら、シミュレーションを実行します。

Studio は現在のところ3種類のデータセットを準備して、モデルの外との間でデータの直接交換を可能としています。それは、マイクロソフト Excel、SAP 社の NetWeaver BI (Business Intelligence) および Studio 独自のデータセットです。

利用頻度が最も高い Excel の利用について具体的に説明するために、前述の簡単なモデルに組み込んで見ましょう。

(1) モデルのデータの入出力条件

- モデルに「配送リード日数」、需要の「平均」と「標準偏差」を Excel から入力します。
- モデルから「倉庫」(在庫量)を Excel へ出力します。

(2) データの入出力設定の基本プロセス

モデルの変数は Excel 上のデータと Excel データセットの変数を介して接続します。

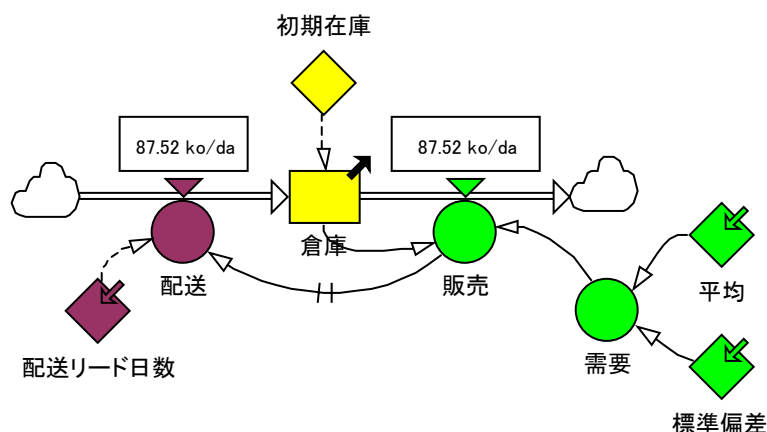
モデルの変数 ⇔ データセットの変数 ⇔ Excel 上のデータ

- ① モデル要素の入出力の定義
- ② モデルの外のデータセット(Excel)の準備
- ③ Spreadsheet Connection Wizard により Excel データセットを準備・設定
- ④ Connections においてモデル変数とデータセット変数の接続を設定

(3) モデルのインターフェース変数の設定

Excel との入出力が予定されているモデルの変数のプロパティ画面を開きます。

プロパティ画面の Advanced タブのページを開き、Connection における Transfer direction を none から in あるいは out に変更します。変更が終わると変数のシンボルに入出力を意味する矢印が付きます。



(4) Excel の準備

Excel データセットは、モデルへの入力データ用の ds-in とモデルからの出力データ用の ds-out の2種類を準備します。それぞれを、販売入出力.xls の Excel ファイルのタブ名 in とタブ名 out に接続させることにします。したがって、販売入出力.xls のタブ名 in には以下を入力データを準備しますが、タブ名 out にはモデルから出力されるのでデータの準備は要りません。

	A	B	C	D
1	time	配送リード日数	平均	標準偏差
2	2007/1/1	3	100	0.2
3				
4				
5				

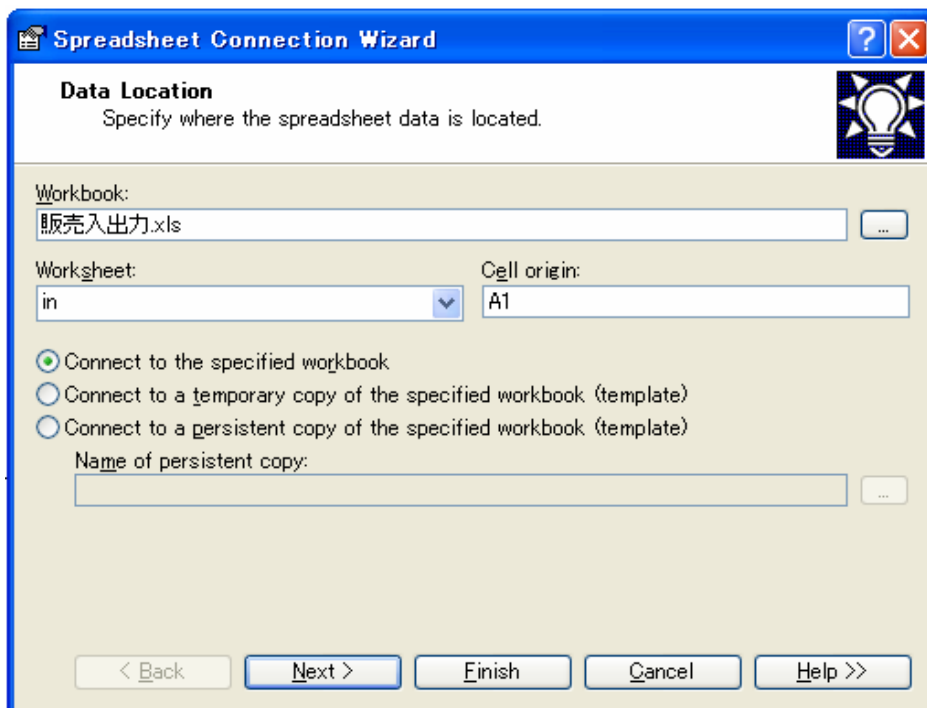
(5) Spreadsheet Connection Wizard によるデータセットの準備

モデルを描いているダイアグラムの空欄で右クリックし、プルダウンメニューから次に示す選択で、

Spreadsheet Connection Wizard を表示させます。

Connect to Dataset ⇒ New Dataset ⇒ Spreadsheet Dataset

このダイアログ・ボックスはデータセット ds-in 用の最初の画面です。



一つのデータセットについて、ページを送りながら、以下の設定をします。

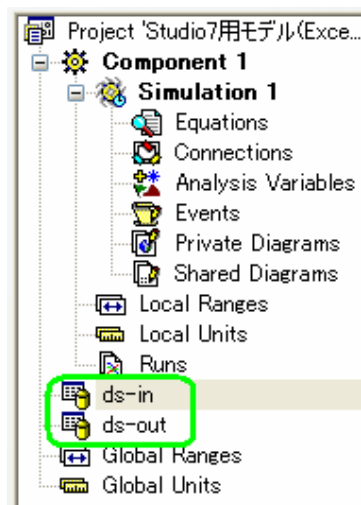
- ▼Data Location(データの保管場所など)
- ▼Time Dependency(時刻ごとのデータか?)
- ▼Misc(入出力頻度などその他のデータの設定)
- ▼Common Dimension(データセットの変数は共通の次元か?)
- ▼Layout(Excel 上のデータの並び方)
- ▼Dataset Variable(データセットの変数一覧)
- ▼Set Dataset Name and Finish(データセットの名前と設定場所の定義 ⇒ 終了ボタン)

ds-in の設定が終わったら、次に、出力用のデータセット ds-out の設定を行います。

その結果、プロジェクト・ウィンドウには、右図のように2種類のデータセットが準備されたことが表示されます。

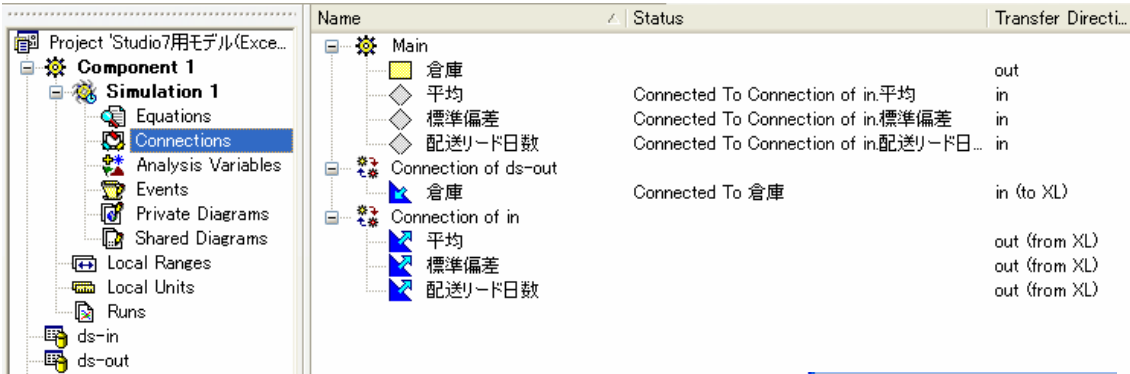
モデルの入出力シンボルの定義が終わり、モデルへ入力する Excel 上のデータの準備と Excel Book の準備も終わり、入力と出力の Excel データセットも準備しました。

残っているのは、モデルの変数とデータセットの変数の接続の定義です。



(6) データセットとモデル変数の連結

プロジェクト・ウィンドウの Connection (Studio 2005 の Co-model) を開き、モデル要素を対応するデータセットにプル・ダウンしてドロップします。下の画面は設定が終わった画面です。



(7) シミュレーション実行

シミュレーションを実行します。

倉庫の在庫量の出力を 5 日刻みで表示した結果を右側に示します。

	A	B
1	time	倉庫
2		ko
3	1/1	200
4	1/6	118
5	1/11	141
6	1/16	66
7	1/21	130
8	1/26	126
9	2/1	106
10	2/5	118
11	2/10	165
12	2/15	168
13	2/19	110
14	2/24	145
15	3/1	162
16	3/6	152
17	3/11	85
18	3/16	97
19	3/21	148
20	3/26	114
21	4/1	103
22		

6.5 戦略シミュレーション機能

(1) 条件設定の仕組み

戦略シミュレーション機能として、最適化、リスク評価、リスク管理の3種類の機能が準備されています。これらはモデルと一体となって、同じ画面上で操作できますから、モデルを構築することと、そのモデルの挙動を戦略的に評価することとを連続して相互に影響を確認しながら進めることができます。

これらの 3 種類の機能を実施するために、シミュレーションを

実行する環境が 2 種類あります。一つは、「一般シミュレーション環境」で、モデルの初期値を与えてその後の挙動を計算します。もう一つは、「確率シミュレーション環境」です。確率シミュレーションのサンプリング法としては、基本的なモンテカルロ法と実験計画法を組み込んでサンプリングの効率化を図ったラティン・ハイパーキューブ法が準備されています。これら一般と確率の両シミュレーション環境には、“Analysis Variables”と名付けられているシミュレーション条件設定画面があります。


戦略シミュレーションを実施するためにはその上で、モデルを構成している変数を、必要に応じて仮定変数 (Assumptions)、決定変数 (Decisions)、目的変数 (Objectives)、影響変数 (Effects) として対応付けます。3 種類の戦略シミュレーションで使用するシミュレーション環境の種類と、上記の 4 つの変数の内で条件を設定すべき項目について一覧表で示します。

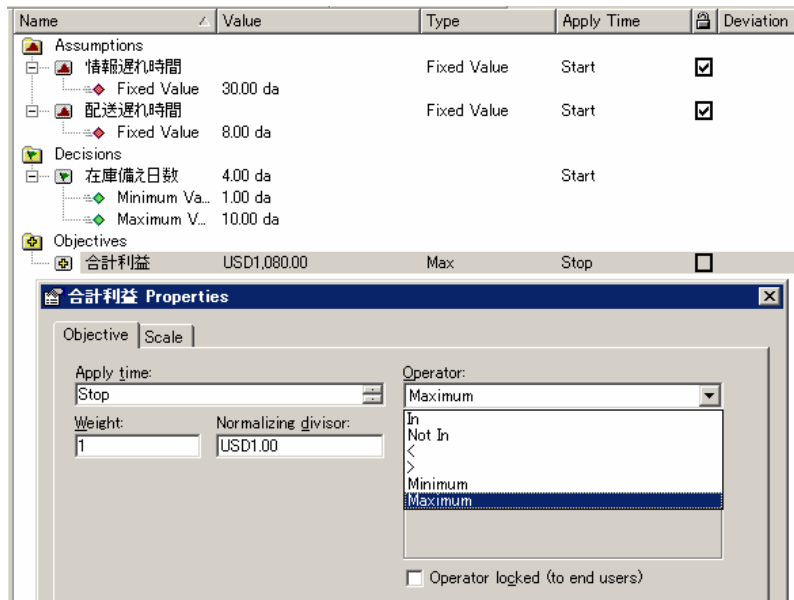
	最適化	リスク評価	リスク管理
仮定変数 (Assumptions)	●	●	●
決定変数 (Decisions)	●		●
目的変数 (Objectives)	●		●
影響変数 (Effectives)		●	●
一般シミュレーション環境 (Simulation)	●		
確率シミュレーション環境 (Risk Analysis)		●	●

(2)最適化の機能

最適化としては、目的変数が「ある範囲内に存在する」、「ある範囲内に存在しない」、「ある値より大きい」、「ある値より小さい」、「最小」、「最大」の6種類があります。

一般シミュレーション環境における Analysis Variables 画面において、シミュレーションの前提条件である Assumption を設定し、次に、最適化を目指して決定される決定変数を設定し、最後に最適化対象の目的変数を設定します。いずれの変数も複数個を対応つけることができます。

その後で、最適化アイコン  をクリックして最適化実行条件を設定した後、最適化を実行します。Studio の最適化では、遺伝的アルゴリズムを応用しています。



曖昧な外生変数を実績データから逆に決定するためのチューニングについては、その実績データとそれに対応するシミュレーション結果との差の2乗和が最小になるように最適化を実行して、曖昧な外生変数を同定します。

(3) リスク評価の機能

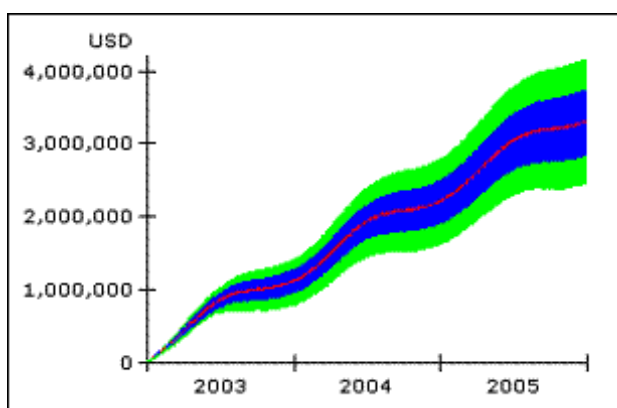
対象としている組織に不明確な外部要因、例えばインフレ率あるいは決定するのが困難な値など、があったとして、それがシミュレーション結果に深刻な影響を与えるのであるなら、その外部要因はリスク要素であると考えられます。

Name	Value	Type	Apply Time
Assumptions			
在庫備え日数		Fixed Value	Start
Fixed Value	2.62 da		
在庫費係数		Uniform	Start
Minimum	5.00 %/da		
Maximum	10.00 %/da		
情報遅れ時間		Triangular	Start
Minimum	27.00 da		
Maximum	35.00 da		
Peak	30.00 da		
配送遅れ時間		Truncated Normal	Start
Expected V...	8.00 da		
Standard De...	2.00 da		
Lower Limit	0.00 da		
Upper Limit	+? da		
Decisions			
Objectives			
Effects			
合計利益		First	
Average	USD3,742,572.69		
Standard De...	USD214,767.80		
10 Percentile	USD3,442,629.99		
25 Percentile	USD3,569,359.20		
75 Percentile	USD3,914,170.37		
90 Percentile	USD4,038,452.12		

Studio のリスク評価の機能を使って、ある仮説に基づき構築されたモデル構造を前提として、リスク要素の変化が最終結果にどのように影響するかについて調査することができます。そのようなシミュレーション結果は、特定の結果を達成できる尤度(ゆうど:ありそうな確率)と呼ばれます。また、この感度分析を使って、最終結果(評価対象の実績)を改善できる支点(レバレッジ)となる変数を見出すこともできます。

結局、リスク評価とは、外生変数として与えられる仮定変数がある確率分布をもった乱数で与えられるとき、モデルで注目しているある内生変数(影響変数)がどのような確率変数として挙動するかを評価することです。上の図は確率シミュレーション環境で開いた Analysis Variables の画面の例です。仮定変数は、一定値のほか、

正規分布、切断正規分布、三角分布、一様分布、指数分布で定義できます。結果としての影響変数は、平均値と標準偏差のほか、特定の%の確率に対応する影響変数の計算値です。サンプリング方法としては、モンテカルロ法とラティン・ハイパーキューブ法を準備していますが、ここでは計算効率の良いラティン・ハイパ



一キューブ法を推奨しています。

上右側の説明図は、リスク評価分析の成果がどんな形で見えるかを示した例です。

(4) リスク管理の機能

前述のリスク評価の場合と同じように、リスク要素である外部要因が想定される場合には、企業としては最終的な年度末のフリー・キャッシュ・フローをある値以上に何%以上の確率で確保したいので、操作可能な変数をどのように決定すれば良いかなどを検討します。

あるいは、操作可能な変数を最適化することにより、何%の確率で確保できるフリー・キャッシュ・フローの値はいくらになるのかを検討します。

このように、リスク下で最適化の機能を実行することをリスク管理と呼びます。リスク管理は、(2)の最適化と(3)のリスク評価とを組み合わせた機能です。上の図は確率シミュレーション環境で開いた Analysis Variables の画面の例ですが、最適化が実行された後のものです。

リスク管理の操作では、まず、Analysis Variables 画面で4種類の全ての変数に対して必要なモデルの要素を対応付けます。その後で最適化アイコンをクリックして、最適化実行条件を設定した後、最適化を実行します。最適化の内容は、(2)最適化の機能の場合と同じです。

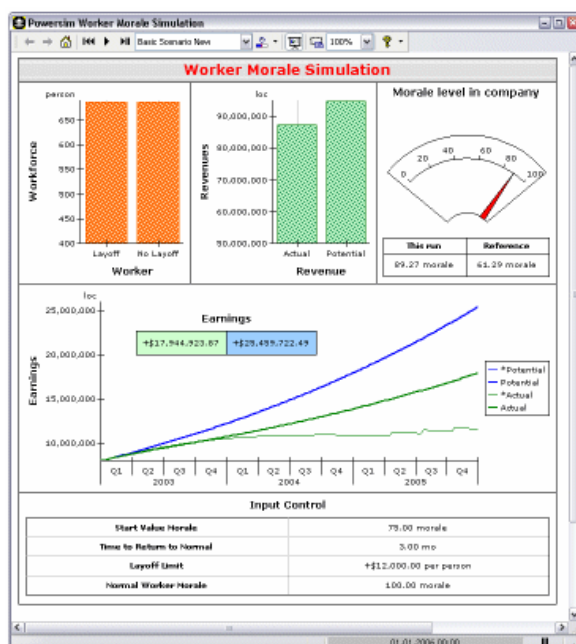
Name	Value	Type	Apply Time	Deviation
Assumptions				
在庫係数		Uniform	Start	<input type="checkbox"/>
Minimum	5.00 %/da			
Maximum	10.00 %/da			
情報遅れ時間		Triangular	Start	<input type="checkbox"/>
Minimum	27.00 da			
Maximum	35.00 da			
Peak	30.00 da			
配送遅れ時間		Truncated Normal	Start	<input type="checkbox"/>
Expected V...	8.00 da			
Standard De...	2.00 da			
Lower Limit	0.00 da			
Upper Limit	+? da			
Decisions				
在庫備え日数	2.28 da		Start	
Minimum Va...	1.00 da			
Maximum V...	10.00 da			
Objectives				
合計利益	90.00 %	>	Stop	<input type="checkbox"/> 0.00 %
Limit	USD3,300,000.00			
Effects				
合計利益				
Average	USD3,552,074.99			
10 Percentile	USD3,285,793.82			
25 Percentile	USD3,373,868.22			
75 Percentile	USD3,698,958.54			
90 Percentile	USD3,799,441.66			

6.6 シミュレーターの作成

Studio は、強力な表示機能、データ入力機能、ブックマークとハイパー・リンク機能、イメージ表示機能などが組み合わせられてシミュレーターになります。モデルを構築するデザインモードからプレゼンテーションモードに切り替えることにより、Studio はシミュレーターに変身するのです。ユーザーはインターネットのブラウザを操作するのと同じ感覚と同じ方法で、シミュレーションを実行できます。もし必要なら、モデルをプレゼンテーションモードで常に開くこともできますし、エンド・ユーザーにはモデルの構造を見えないようにすることもできます。また、パスワードのプロテクトを各種のアクセスレベルで設定できます。

Studio はシミュレーション・モデルの中で特定の変数の値が設定条件と一致したときに、ある特定の操作ができるように、イベントとアクションの機能が用意されています。イベント機能を使うと、設定したパラメータの値に到達したら、小さなウィンドウをポップアップしてその中にメッセージを表示するか、あるいはハイパー・リンクを働かせることができます。

下の図は、シミュレーションブラウザで示されたシミュレーターのインターフェースの一例です。



6.7 モデルの再利用を可能にした階層構造

Studio では階層的にモデルを築くことができます。下位のモデルをサブモデルと呼びます。サブモデルの作り方は2通りあります。一つは、モデルが複雑になるのを防ぐために、モデルの中のセクターと呼ばれるようなまとまった塊を、メインモデルの中に新たに配置したサブモデルのダイアグラムの上で構築する方法です。もう一つは、以前作成していた既存のモデルを、新しいプロジェクトのメインモデルの中で、サブモデルとして引用して挿入するやり方です。

いずれも大型の複数モデルによるプロジェクトを構築するには必要不可欠な機能です。特に、後者が整備されたことで、応用機能別のモデル・ライブラリを準備してそれを組み合わせて利用し、効率的に大型のモデルやその集合体であるプロジェクトを完成させることが可能になりました。このことは、SD の実用化の面において大変大きな進歩をもたらすことになりました。

また、シミュレーション・モデルの中でサブモデルを使うことにより、シミュレーションの利用者から不必要で詳細すぎる部分を隠すことができますから、モデルの表現自身が大変分かりやすくなります。

6.8 先進的な機能

Studio は上述以外にさまざまな先進的なモデリングの特徴を持っています。その特徴は組織あるいは問題のシミュレーション・モデルを作り出す上で大いに役立ちます。今までの説明では取り上

げなかったいくつかの重要な機能を簡単に説明します。

モデルの一貫性のチェック :Studio はモデルが矛盾を含んでいる場合に、方程式においてもモデルの構造を示すダイアグラムにおいても警告を発します。

配列 :Studio は配列変数に特徴があります。配列はモデルの中で層(レイヤー)を作るために使われます。これによりモデルの中でモデルの構造を繰り返し作成する無駄な工数がなくなります。例として、工場が 5 種類の製品を同じ製法で作っていたとします。そうすると 5 要素の配列を含んだプロセス・モデルを作ることになるでしょう。これらの配列で使われる要素のインデックスはレンジ設定において、それらを簡単に区分けできるように特徴のある名前にするか、またはよくあるように整数のインデックスをつけるかどちらかです。これらの要素の定義は、それぞれのシミュレーション・プロジェクトの中で保持されます。そして、同じプロジェクトの中では、すべてのシミュレーション・モデルにおいてこの定義が使われることになります。

複数のコンポーネントとシミュレーション :Studio ではそれぞれのシミュレーション・プロジェクトに、必要な数だけコンポーネント(モデル)を含めることができます。コンポーネントとは自己完結型のシミュレーション・モデルのことです。シミュレーション・プロジェクトはグローバル単位とグローバル・レンジの定義を持っています。このグローバルな定義はプロジェクトに含まれるすべてのコンポーネントにおいて使われます。さらに、それぞれのコンポーネントは複数のシミュレーションを持っています。このシミュレーションの一つとしてリスク評価の課題も含まれることになります。それぞれのシミュレーションは、それぞれの独自の I/O 操作画面、時間条件、イベントとアクション、データセットの接続の設定を持っています。

複数の積分法 :Studio は複数の積分法を持っています。オイラー法を利用するのが基本ですが、もっと正確な計算を確実にするためには、高次の積分法を選択することもできます。それ用の積分法が 2 次、3 次、4 次のルンゲ・クッタ積分法です。

離散、連続、論理フロー :Studio ではさまざまな積分技術を使ったフローを実装することができます。一般の連続流以外に、離散流と論理流があります。

7. 技術サポート

Powersim 社と POSY 社とは異なるレベルの各種サポートを提供しています。

(1) 独習の続き

POSY 社のホームページの左側のフレームにある“Ps Studio マニュアル”には、本説明書のアップグレードした内容が掲載されています。

同じく“テスト用モデル ダウンロード”を開いてください。その中に“システムダイナミクス学習向けモデル”があり、そこに次の二つのモデルが掲載されています。このモデルをダウンロードして、システムダイナミクスと Ps Studio の学習の続きにお使い下さい。

▼SD と Studio 操作法の基礎

▼モデリングの基礎

POSY 社の HP の URL <http://www.posy.co.jp>



(2) オンライン・ヘルプ

問題を解決するための最初の取り組みとして、Studio のすべてのバージョンに含まれているオンライン・ヘルプを検索してください。もし、ヘルプ・システムの中で見つけた情報を使っても問題を解決できない場合には、Powersim Software AS のサポート・チームが支援します。

(3) マニュアル

Studi7のHelpから、「目次」のタブを開いてください。目次に沿った約 850 ページ程度のマニュアルとしてご利用いただけます。

印刷する場合には、その項目のサブ・トピックスまでを、まとめて印刷できます。

(4) Online support request form を通したサポート

支援を受けるには on-line support form を通して技術支援要望のフォームを Powersim 社にメールでを送るか、または、電話をかけてください。ただし、その前に、Powersim 社から Expert 版以上の Studio を購入し、保守契約(SUA: Service and Up Grade Agreement)を結んでおく必要があります。

Powersim 社のサポート案内の URL <http://www.powersim.com/support/default.asp>

(5) POSY 社の個別対応

Powersim グループとして正式な対応ではありませんが、POSY 社では時間が長くかからない問題に関しては極力相談に応じています。

特に、オリエンテーション・コースを受講されて、その後、自組織のモデルを構築しようとしている方々については、時間が許す限り優先的に対応させていただきます。

以上

連絡先 : POSY Corp. 〒102-0092 東京都千代田区隼町 2-12-104 藤和半蔵門コープ1F

<http://www.posy.co.jp>

matsu@posy.co.jp

松本 憲洋 電話 03-3512-5358 080-5047-3849



【利用頻度が高い関数の概説 一覧表】

論理関数

A > B	超過	AはBを越えている。
A >= B	以上	AはB以上である。
A < B	未満	AはB未満である。
A <= B	以下	AはB以下である。
A <> B	不等価	AはBと等しくない。
A ANB B	論理積	条件Aでありかつ条件Bである。

操作記号

A + B	加算	AにBを加える。
A - B	減算	AからBを引く。
A * B	乗算	AにBを掛ける。
A / B	除算	AをBで割る。
A ^ B	べき乗	AにB乗する。
A = B	等号	AはBに等しい。

財務関数

PV(利率, 支払期間回数, 定期支払額, 将来価値)	現在価値
<p>現在価値 (PV : Present Value)、将来価値 (FV : Future Value)、定期支払額 (PMT : Payment) の関係は下式で与えられている。 $PV * (1+r)^n + (PMT/r) * [(1+r)^n - 1] + FV = 0$ ここで、"r"は、支払期間に対応した利率であり、"n"は支払期間回数である。 上式からPVは計算される。 なお、財務関数の約束ごととして、支払金額は負記号、受け取り金額は正記号で表現する。初回の支払は、最初のTimestepの期末に発生する。</p>	
FV(利率, 支払期間回数, 定期支払額, 現在価値)	将来価値
PV(現在価値)関数を参照	
PMT(利率, 支払期間回数, 現在価値, 将来価値)	定期支払額
PV(現在価値)関数を参照	
NPV(定期支払額, 利率)	正味現在価値
<p>投資についてリターンを正味現在価値で評価するために使う。利率は支払期間に対応した利率である。 定期支払額と利率はIF文などによる条件の外部操作により支払期間ごとに変化させることができる。 一般には、t=0における値が投資額となる。 なお、財務関数の約束ごととして、支払金額は負記号、受け取り金額は正記号で表現する。 計算式は下記による。 T=nが計算終了回数で、</p>	
$NPV = \sum_{t=0}^n \frac{PMT_t}{(1+r_t)^t}$ n timestep で与えられる。	

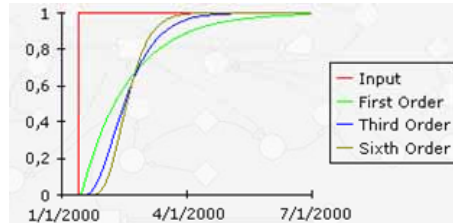
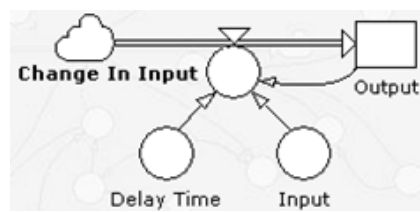
統計関数

MAX(A, B, C, ...)	最大値
入力信号(A, B, C, ...)の内の最大値を出力する。	
MIN(A, B, C, ...)	最小値
入力信号(A, B, C, ...)の内の最小値を出力する。	
RUNAVERAGE(入力値)	全平均
シミュレーションの全区間にわたる入力値の平均を出力する。	
RUNSTDEV(入力値)	移動標準偏差
シミュレーションの全区間にわたる入力値の標準偏差を出力する。	

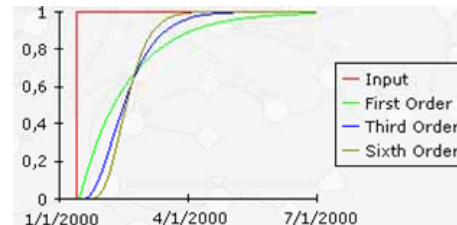
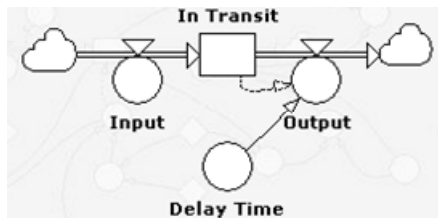
遅れ関数

DELAYPPL(入力信号, 遅れ時間, [初期値])	パイプライン遅れ
<p>入力信号を遅れ時間だけ遅らせて入力信号の形のままで出力する。コンベア遅れとかトコロテン遅れと言われる所以である。</p> <p>シミュレーション開始から遅れ時間の間は、初期値を出力する。</p> <p>初期値は中括弧で囲んで初期値の古い順に{p, q, r, ...}と表す。遅れ時間の間中、同じ値をとる場合には中括弧を書かないで一つの数値を単独に記載する。</p> <p>また、初期値を記載しない場合には、デフォルト値として入力値の初期値が代用される。</p>	

DELAYINF(入力信号, 遅れ時間, [次数], [初期値])	情報遅れ関数
<p>入力信号を指数遅れ関数で変換して出力する。“遅れ時間”が指数遅れの平均遅れ時間である。“次数”は整数値で、入力しない場合はデフォルト値の1次とみなされる。</p> <p>また、初期値を記載しない場合には、デフォルト値として入力値の初期値が適用される。</p> <p>この関数をSD表記すると、左側のモデルとなる。高次の遅れ関数の場合には、同じモデルがカスケードに下方に並び、上段のOutputが下段のInputとなる。</p> <p>遅れ時間35日で、ステップ関数が入力された場合の1次、3次、6次の出力値の例を右図に示す。</p>	



DELAYMTR(入力信号, 遅れ時間, [次数], [初期値])	物質遅れ関数
<p>入力信号を指数遅れ関数で変換して出力する。“遅れ時間”が指数遅れの平均遅れ時間である。“次数”は整数値で、入力しない場合はデフォルト値の1次とみなされる。</p> <p>また、初期値を記載しない場合には、デフォルト値として入力値の初期値が適用される。</p> <p>この関数をSD表記すると、左側のモデルとなる。高次の遅れ関数の場合には、同じモデルがタンデムに右方向に並び、左側のOutputが右側のInputとなる。</p> <p>遅れ時間35日で、ステップ関数が入力された場合の1次、3次、6次の出力値の例を右図に示す。</p>	



次数が大きくなると、delaypplの出力に近づくが、その状況は計算精度により大きく影響を受けるので注意が必要である。

グラフ関数

GRAPH(X, X0, dX, {Y1, Y2, Y3...})	グラフ関数
この関数はX-Yグラフの定義をして、横軸X軸の入力値Xに対するY軸の値を出力する。 グラフの定義では、横軸X軸として初期値X0とその等間隔の増分dXを与え、その各X値に対する縦軸Yの値を、中括弧で囲んで{Y1, Y2, Y3...}のように表現する。 縦軸のデータが存在する範囲では、入力値Xの両側の値を使った直線内挿値を出力し、範囲外の場合にはそれぞれの外端の値を出力する。	
GRAPHLINAS(X, X0, dX, {Y1, Y2, Y3...})	直線グラフ関数
この関数はX-Yグラフの定義をして、横軸X軸の入力値Xに対するY軸の値を出力する。 グラフの定義では、横軸X軸として初期値X0とその等間隔の増分dXを与え、その各X値に対する縦軸Yの値を、中括弧で囲んで{Y1, Y2, Y3...}のように表現する。 縦軸のデータが存在する範囲では、入力値Xの両側の値を使った直線内挿値を出力し、範囲外の場合には入力値Xに近い2点で直線を決め、その線上の値を外挿値として出力する。 すなわち、GRAPH関数との違いは、外挿値の求め方だけである。	
GRAPHSTEP(X, X0, dX, {Y1, Y2, Y3...})	ステップグラフ関
この関数はX-Yグラフの定義をして、横軸X軸の入力値Xに対するY軸の値を出力する。 グラフの定義では、横軸X軸として初期値X0とその等間隔の増分dXを与え、その各X値に対する縦軸Yの値を、中括弧で囲んで{Y1, Y2, Y3...}のように表現する。 縦軸のデータが存在する範囲では、入力値Xの左側の値を内挿値として出力し、範囲外の場合にはそれぞれの外端の値を出力する。	
GRAPHCURVE(X, X0, dX, {Y1, Y2, Y3...})	曲線グラフ関数
この関数はX-Yグラフの定義をして、横軸X軸の入力値Xに対するY軸の値を出力する。 グラフの定義では、横軸X軸として初期値X0とその等間隔の増分dXを与え、その各X値に対する縦軸Yの値を、中括弧で囲んで{Y1, Y2, Y3...}のように表現する。 縦軸のデータが存在する範囲では、与えられた全点を使って3次の多項式補間法で内挿点を計算して出力値とする。したがって、与える(X,Y)の点列が3点の場合には2次式で、2点の場合には1次式で内挿値を計算する。与えられたX軸の範囲外の場合には、GRAPHLINAS関数と同じく、入力値Xに近い2点で直線を決め、その線上の値を外挿値として出力する。	

三角関数

SINWAVE(振幅, 周期, [位相遅れ])	サイン関数
シミュレーションの時点(Time)ごとのサイン関数値を出力する。 周期と位相遅れは時間単位で入力する。	
SINWAVE(振幅, 周期, [位相遅れ])	コサイン関数
シミュレーションの時点(Time)ごとのコサイン関数値を出力する。 周期と位相遅れは時間単位で入力する。	

変換関数

INTEGER(実数の入力値, [分解度], [補正值])	整数化
共に整数の分解度と補正值が入力されない場合には、分解度=1、補正值=0がデフォルト値となり、実数の入力値に対する整数値が出力される。 分解度と補正值が与えられると、次のX値を出力する。 $X = K * \text{分解度} + \text{補正值}$ ただし、K値は、Xの絶対値が実数の入力値の絶対値以下で、かつ最大になる条件を与える整数値。	

ランダム関数

NORMAL(期待値(平均値), 標準偏差, [SEED])	正規分布乱数
---------------------------------------	---------------

統計解析すると次の特性を持った乱数を発生する。発生した乱数は、平均値(期待値)と標準偏差で定義される正規分布となる。

SEEDを与えないと毎回異なる時系列の乱数が発生する。

SEEDに0~1の値を与えると、その数値に相当するパターンの乱数が毎回発生する。

RANDAM(最小値, 最大値, [SEED])	一様分布乱数
---------------------------------	---------------

統計解析すると次の特性を持った乱数を発生する。発生した乱数は、最小値から最大値の間で一様分布をとる。

SEEDを与えないと毎回異なる時系列の乱数が発生する。

SEEDに0~1の値を与えると、その数値に相当するパターンの乱数が毎回発生する。

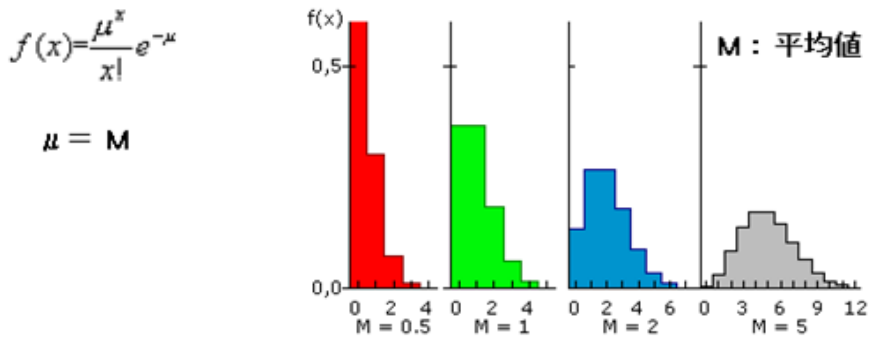
POISSON(期待値(平均値), [SEED])	ポアソン分布乱数
----------------------------------	-----------------

ポアソン分布は時間などのある測定単位で発生する事象数、ある商品の顧客あたりの購入数などの低い確率(希少現象)に対して多量の観測が可能ときに用いるランダム関数である。期待値(平均値)は $N \ll \langle \text{個} / \text{日} \rangle$ のように測定単位に対する値を入力する。出力は離散的に整数で与えられる。

この関数は期待値が大きくなると分布形状がNORMAL関数に似てくる。

SEEDを与えないと毎回異なる時系列の乱数が発生する。

SEEDに0~1の値を与えると、その数値に相当するパターンの乱数が毎回発生する。



時間関連関数

TIMECYCLE(開始時刻, インターバル時間, 継続時間)	繰返時間の設定
--	----------------

開始時刻からインターバルタイムで繰り返し、インターバルタイムの継続時間だけ”真値”を出力し、残りの時間は、”偽値”を出力する。この関数をIF文の中で条件として用いる。

条件関数

IF(条件, 真の場合の出力, 偽の場合の出力)	条件文
---------------------------------	------------

入力された条件に対して真の場合と偽の場合で異なる値を出力する。

条件 : A=B, A>B, A<>B, A<=B, A<B, A<=B, A<>B, A and B

配列関数

FOR(配列の次元1, 配列の次元2, . . .)	配列の要素に対する演算内容)	配列計算の基礎
<p>「配列の次元」の主な表現方法を以下に示す。</p> <p>①数値による表現 numerical subrange 1..5</p> <p>②文字による表現 enumerated range 「関東地域」 「関東地域」はGlobal Rangeで{埼玉, 千葉, 東京, 神奈川}のように定義され</p> <p>③文字による部分表現 enumerated subrange 千葉..神奈川 「千葉..神奈川」は「関東地域」の一部であることをGlobal Rangeで定義する。</p> <p>使い方: FOR(i=1..3,j=1..2 AAA[i]*BBB[j]) --> {{30,40},{20,30},{10,20}} FOR(i=関東地域 CCC[i]*BBB[i]) --> {100,150,500,300}</p>		
ARRSUM(入力される配列)		配列要素の合計
<p>「入力される配列」の要素の合計を計算する。</p> <p>使い方: ARRSUM({4,2,3,6,2,1}<<m>>) --> 18<<m>> ARRSUM({{3,1,2},{9,7,3},{10,2,7}}) --> 44</p>		
CONCAT(配列1, 配列2, . . .)		配列の連結
<p>引数の配列1, 配列2. . . を結合 (concatenate)する。</p> <p>使い方: CONCAT({1,2,3,4},{5}) --> {1,2,3,4,5} CONCAT({{1,2},{3,4},{5,6}}, {{7,8},{9,10}}) --> {{1,2},{3,4},{5,6},{7,8},{9,10}} CONCAT(FOR(i=1..1 A[i]/2), FOR(i=2..6 A[i]*2)) --> {5,40,60,80,100,120}<<m>> ここで、A = {10,20,30,40,50,60}<<m>></p>		
INDEX(整数)		指数化
<p>配列の要素を表示する場合に、配列の要素番号を指定する。 要素番号を指数と言ひ、普通の数字を指数(要素番号)に変換する関数がINDEX関数である。</p> <p>使い方: 配列XX= {{3,2,5,3},{6,3,4,1}} A = 2 . . . 整数 B = 3 . . . 実数 配列XX[Index(A)] --> {6,3,4,1} 配列XX[Index(A),Index(INTEGER(B))] --> 4</p>		
LOOKUP(入力される配列, INDEX)		配列要素の表示
<p>配列の中の要素を要素番号であるINDEXで指定して表示する関数である。 要素番号は、「何番目の要素」を意味している。 この関数の結果は、「入力される配列[Index]」となる。</p> <p>使い方: 配列YY = {{1,2,3,4},{5,6,7,8}}とする。 LOOKUP(配列YY, 1) --> {1,2,3,4} LOOKUP(配列YY, 2) --> {5,6,7,8} LOOKUP(配列YY,1,2) = 2 LOOKUP(配列YY,2,1) = 5</p>		

NUMERICAL(レンジの要素名)	数値化
<p>配列の次元を文字で表現することができる。 cf. FOR関数 たとえば、「関東地域」というレンジの要素は、埼玉, 千葉, 東京, 神奈川である。 配列計算の中では、この要素を文字ではなく整数で表現する必要が出て来る。 そのような場合に、“1”から始まる整数でレンジの要素を番号に変換するときに使う関数である。 使い方: 埼玉, 千葉, 東京, 神奈川の要素を持つ関東地域というレンジを考える。 NUMERICAL(埼玉) = 1 NUMERICAL(東京) = 3</p>	
SCANEQ(ベクトル配列, 指定する数値[, Last=False [, 個数])	合致する要素番号
<p>ベクトルの配列の要素が、指定する数値に一致する最初の要素番号を求めるときに使う関数である。 但し、Last=Trueとした場合には、Lastすなわち最後に一致した要素番号が結果となる。 複数の要素が一致する場合に複数個の要素番号を求めるには、個数の位置にその数を入力する。 一致する要素がない場合には“0”が表示される。 使い方: A = {5,4,1,3,6,3,4,8,1,2,4,2,6,4,9}のベクトルを考える。 SCANEQ(A, 4) --> 2 SCANEQ(A,4,TRUE) --> 14 SCANEQ(A,4, FALSE, 3) --> {2,7,11}</p>	
TRANSPOSE(マトリックス配列)	配列の転置
<p>マトリックスの行と列とを入れ替える転置を行う関数である。 使い方: MatrixA = {{1,2,3}, {4,5,6}, {7,8,9}} を転置する。 TRANSPOSE(MatrixA) = {{1,4,7}, {2,5,8}, {3,6,9}} MatrixB = {{1,2}, {3,4}, {5,6}, {7,8}, {9,10}} を転置する。 TRANSPOSE(MatrixB) = {{1,3,5,7,9}, {2,4,6,8,10}}</p>	