

# はじめての “Powersim Studio”

2011 年 6 月 15 日

POSY Corporation

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

松本 憲洋

[nmatsumoto@posy.co.jp](mailto:nmatsumoto@posy.co.jp)

## Contents

1. このテキストの狙い	..... 3
2. システムズ・アプローチ	..... 4
2. 1 システム的な思考法の必要性	
2. 2 企業経営におけるシステムズ・アプローチ	
2. 3 システムズ・シンキングとシステム・ダイナミックス	
3. モデリングの基礎	.....11
3. 1 範囲の設定	
3. 2 問題の定義	
3. 3 フィードバック・ループ	
3. 4 レベルとフロー	
3. 5 遅延	
4. Powersim Studio の操作の初歩	.....16
4. 1 Powersim Studio の種類	
4. 2 最初の画面	
4. 3 メニューバーとツールバーでよく使うアイコン	
4. 4 Help 機能の使い方	
4. 5 変数のシンボルと入出力表示部の属性の定義	
4. 6 単位の設定	
5. モデルの構築	.....26
5. 1 問題の定義	
5. 2 モデルの範囲の決定	
5. 3 時系列挙動図ほかデータの収集	
5. 4 要素の抽出	
5. 5 概念モデルの設計	
5. 6 モデルの完成	
5. 7 シミュレーション	
附録1 :定性的なシステムズ・アプローチ(システムズ・シンキング)	32
附録2 :Ps Studio 8 Demo/Express	.....38

## 1. このテキストの狙い

我々は日常、様々な問題に遭遇しています。世界や日本における大きな問題、地域社会における問題、かかわっている仕事における問題、自分や家庭における問題、などなど大小様々な問題があります。我々はそれらを解決する必要があります。

簡単な問題は自分の頭の中のメンタル・モデルを駆使して解決してきました。しかし、インターネット・ウェブが地球を覆い、遠く離れていると思っていた他国の動向が、瞬時にして日本に影響するグローバル社会では、自分がかかわっている問題を世界から切り離して考えることができなくなってきました。このような状況では、過去の経験に基づいている自分の頭の中のメンタル・モデルだけに、問題解決を頼っていたのでは、解決できない、あるいは解決を誤ってしまうような問題が多くなってきたように思います。システム・ダイナミクスは、そのような問題をより広い環境の中で取り扱い、全体的に眺めて解決するための方法論です。

このテキストは、システム・ダイナミクスに基づくモデリングとシミュレーションを始める人向けに書いたもので、そのために準備されているツールである“Powersim Studio”を使い始める第一歩をサポートします。しかし、具体的なツールの操作の説明に入る前に、この第1章と第2章とで、システム・ダイナミックの概要について簡単にお話ししましょう。

古くはギリシャ時代から、問題が発生するとその問題の原因を探するために、対象をツリー構造形式に分解し、その各要素を分析することで、原因を探りあて、その問題を解決することが科学的であると考えられてきました。これを、要素還元法と呼んでいます。

しかし、1900年に入ってから、例えば生物学では、動物の体の心臓とか胃などの各要素の機能を個別に調べてその機能を足し合わせただけでは、その動物全体の機能を説明できない、すなわち、要素の機能だけでなく、要素と要素との関係性まで含めて足し合わさないと全体の機能を説明できないことが指摘され、システム論が現れました。同じ動きは制御工学においてもあり、例えばウイナー(Norbert Wiener)により著されたサイバネティクスはその代表例として挙げるすることができます。

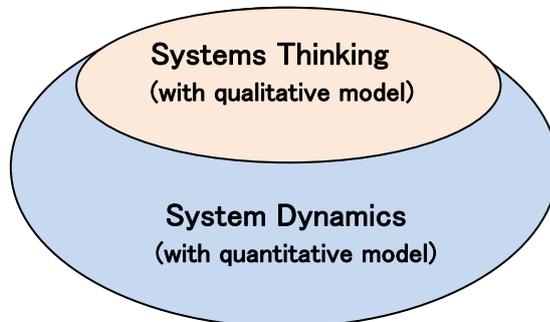
対象に関らず問題の解決に向かってこのようにアプローチするやり方は、“システムズ・アプローチ”と呼ばれています。システムズ・アプローチは、“複数の要素が、秩序を維持するための前提条件を互いに供給し合う関係を、筋道を立てて論理的に、手順を追って把握する、体系的で秩序だったアプローチ”と定義することができます。

MITスローン・スクールのフォレスター(J.W.Forrester)教授が1956年に創案したシステム・ダイナミクスもシステムズ・アプローチとして分類されます。このシステム・ダイナミクスは様々な名前で呼ばれてきました。Industrial Dynamics、Urban Dynamics、World Dynamics、Business Dynamics などです。これらを総称してシステム・ダイナミクスと呼んでいます。

話は少々ややこしいのですが、このシステム・ダイナミクスの中には、定性モデルを取り扱う方法論である Systems Thinking(システムズ・シンキング)と定量モデルを取り扱う System Dynamics(システム・ダイナミクス)とがあります。

前述の“システム・ダイナミクス”は、いわばフォレスターによるシステムズ・アプローチのことで、

広義のシステム・ダイナミックとも呼べるでしょう。そうすると、後者のシステム・ダイナミクスは、狭義のシステム・ダイナミクスと呼ぶことになります。両者の関係は下図のように表すことができます。



フォレスターによるシステムズ・アプローチ  
(広義のシステム・ダイナミクス)

さて、システム・ダイナミクスは、経営者あるいはそのスタッフが複雑な課題を解決するための道具として開発されたモデリングとコンピュータ・シミュレーションの技法でしたから、第一のターゲット・ユーザーは経営者あるいはそのスタッフでした。現在では適用範囲が広がって歴史や文学から生物学、物理学、経済学にいたる、ありとあらゆる分野で使用されています。

システム・ダイナミクスに基づくモデリングとシミュレーションを習熟するにはある程度の期間が必要ですが、このテキストでは基本的なテクニックや考え方に限って解説して、初歩をマスターするための必要時間を短縮したいと思っています。このテキストを通じて読者は人間がどうやって問題解決を行うのか、システムに関する簡単な理解がいかに問題解決のスキルを向上させるのか、そして、なぜ我々自身の考え方が変わる必要があるのかということを学んでいきます。このテキストを読み終わるとシステム・ダイナミクスに関する基本的な理解が得られ、Powersim を使う準備が整うことになります。また、初心者にとってこのテキストはモデリングそのもののクイック・スタートの良き手助けとなることでしょう。

## 2. システムズ・アプローチ

### 2.1 システム的な思考法の必要性

伝統的な思考法である要素還元法に則って、小さい部分に分解するという問題解決方法は、我々が普段悩まされている様々な事象をコントロールする上で大変に役だっています。仮に、この能力が我々に備わっていなければ、ほんの些細な問題も実際以上にはるかに大きな問題に見えてしまうはずです。

この“分解して精査する”という問題解決方法が、役に立つ領域があることはだれもが認めるころではありますが、この方法は不十分だったり、時には危険であったりもします。企業の問題をこのような考え方で分析する場合にも重大な誤りに陥る可能性があります。

企業は組織的には経理部、営業部、人事部などの部品に分かれています。企業において何か問題が起きた時に、要素還元法に沿った考え方で分析を進めると、企業全体に影響する問題も、あ

る一つの部門に調査や非難が集中することがよくあります。

例えばマーケットシェアの低下の際に営業部が調査の対象になったり、さらに非難されもするので、これは営業マンが売らないからシェアが低下したと短絡的に考えられてしまうからですが、この図式で見落とされているのは、営業部が活動するためには様々な部門が関係しているという事実です。

売上げが落ちるのはそれらの関連部門のせいだったり、すべての部門のせいだったりすることがあります。あるいは、情報システム部が営業部に必要なコンピュータ支援を提供していないのかも知れないし、いかに受注の管理によりバックログが発生してすぐに商品が欲しい顧客のニーズに応えられていないというような製造部門が問題なのかも知れないのです。このように、多くの要素が問題の原因である可能性があり、企業の各部門の相互関係を分析して始めて明らかになる場合が多いのです。

システムとは、“複数の要素が、秩序を維持するための前提条件を互いに供給し合う関係の集合体”のことです。したがって、我々が周りに存在する要素が集って構成されたグループを眺めたり、要素間の相互関係を眺めているとき、我々はシステムを眺めていることとなります。

循環器系や神経系のような肉体的なシステム、政治や経済といった社会システムなどと共に、企業もいくつもの部門からなりたったシステムであり、その各部門自身もまたシステムとして動いています。要素とその相互関係を分析するというのは、システムを全体として分析することにほかなりません。

対象となるシステムの主要な構成要素をリストアップして、それら要素間の因果関係を表示する関係図を“因果関係図”(Causal Loop Diagram)と呼んでいます。システムズ・アプローチでは後述の“時系列挙動図”(Reference Mode)と共に、因果関係図は定性的モデリング手法の重要なツールです。

さらに考察を進めると、システムに関する一般的な知識あるいは理論は、対象を特定せず種々の対象のシステムに応用できるということが推測できます。すなわち、システム論は対象を特定化しないで汎化した形で、システム全体を理解するために活用できるのです。これについては、センゲ(Peter M. Senge)のThe Fifth Discipline(邦訳:最強組織の法則)とそれに続くField Bookを参照してください。

伝統的なものの見方や考え方を変える必要があるのには理由がいくつかあります。全てのものをこまかく分解して理解しようとする考え方は、必ずしもすべての場合に最適であるとは限らないのです。システムの要素間の結びつきや相互作用のほうがより重要な場合があるのです。システム論はシステム全体を学ぶことにより複雑な問題を解決することを学ぶことであります。そのための方法論としてフォレストーによるシステムズ・アプローチ、すなわち具体的にはシステムズ・シンキングとシステム・ダイナミクスが存在し、有効に活用できます。

## 2.2 企業経営におけるシステムズ・アプローチ

グローバルな環境の中で、企業は多くの試練にさらされ、企業間の競争はかつてないほど複雑な

ものになっています。激しく変わる世の中では、どんなにうまくいっている企業でも変化に対して十分対応するだけのものを持っているとは言えない状況です。企業はこの混沌の中でダイナミックに変化する問題を掌握し、絶えず発生する問題をタイムリーに解決していかなければなりません。また、変化によって必然的に生じるシステムの不安定性にも目を向ける必要があります。

「ダイナミック」という言葉は変わり続けるということを意味しています。昨日と今日とでは状態が異なり、明日になればさらにまた変わっているわけで、システムが破壊されない限り、変化が止まることはありません。常に変化している状況下で、企業はどうやって危機を乗り越え、存続のための大きな決断をすることができるのでしょうか。残念ながら企業の経営者は我々と同じように、場当たり主義的に行動することが多いのではないのでしょうか。問題が起きてからそれに対処し、次が起ればまたそれにといった具合です。一步下がって全体を見て、部分的な事象ではなく全体の挙動のパターンを読み取ろうというようなことは、だんだん難しくなっているように思われます。

システムズ・アプローチでは、このようなシステムの構造やそこに現れる挙動の全体パターン、さらには様々なシステムに共通する挙動をとらえるところから、アプローチを始めます。したがって、システムズ・アプローチを学ぶことによって身につける経営技術の中で重要なのは、個々のイベントを見出し分析することではなく、対象システムの全体の挙動の中に流れているパターンを見出す能力なのです。その挙動はシステムの構造に基づくものだということも学ぶことになります。

我々は一般には、例えば、マーケットシェアの喪失のような状況に臨むと、まずはそれに特化して反応するようにできています。このような場当たり主義のアプローチではその結果として、個人やどこかの組織を非難することになるでしょう。例えば、営業担当の副社長や営業部などがそのターゲットになるのです。

しかし、挙動のパターンを見出すことができる者なら、20年間で4回のマーケットシェアの浮き沈みがあったとするなら、商品投入履歴からそれらが企業の主力商品の交代時期と一致していることを発見することができるでしょう。このように対象システムの全体の挙動の中に流れているパターンを見出すという考え方は、場当たりに問題に直近している部分のみを見る考え方より、はるかに多くの問題解決の選択肢を与えてくれます。

しかし、場当たり主義の意思決定からパターン分析に基づく意思決定へというだけでは十分ではありません。次のステップは、構造と挙動の概念をマスターすることです。システムの挙動、さらにはシステムの健全性(安定性)は、信頼できるデータによって確かめることができます。

例えば、売上、支出、バックログ、配達の遅延などのデータのことですが、この種のデータは毎日経営者に届けられるはずですが、これらのデータを時間軸の上にプロットしたものは、“時系列挙動図”(Reference Mode)と呼ばれています。時系列挙動図は具体的なデータが存在しなくても、その専門家、企業経営の場合には経営者あるいは経営スタッフ、のメンタル・モデルに基づいて描くことができます。

全体像の把握に加えて時系列挙動図を得ることが、次のステップである、どうやって事象は発生するのかについての分析を助けてくれます。

対象システムを構成する要素間の因果関係は分かっているにしても、それらの要素変数がどのよ

うに結び付いているのか、相互にどのように影響しあっているのかを知る必要があります。その際、時系列挙動図にプロットされたデータからは様々な数値の組み合わせが考えられます。

例えば、売上げが伸びたのに利益が減ったのは、収入の伸びを支出の伸びが上回ったからかもしれません。また逆に、規模の経済の結果として支出が減るとすれば、売上げの伸びは支出の減少を引き起こすはずで、これらの事象を説明するものはシステムの構造に他なりません。構造を変えることが現象を変えることにつながるのです。

システムズ・アプローチの基礎にあるのは、すべてのダイナミックな現象は構造に由来するという考え方です。これまで、システムの要素とその相互作用、全体像の把握のための時系列挙動図の利用、そして要素がどのように影響しあうかということについて述べてきました。この“構造”という言葉がこのテキストを通してシステムの要素がどのように統合されているか、すなわちお互いにどのように結び付いているかという意味で用いることにします。時系列挙動図の話からもわかるように、挙動とは構造から直接導かれるものなのです。

これはシステムズ・アプローチの中で非常に重要な概念です。というのはシステムの構造が成功や失敗を規定するものだからです。問題の解決はシステムの中に存在するのだということを理解する必要があります。しかしながら、これは企業の問題はすべて企業の内部に存在するという意味するものではありません。システムは大きなシステムのサブシステムであることが多いということをお出ししてください。そして大きなシステムはさらにまた大きなシステムのサブシステムであることもあります。企業は自分のポリシーから引き起こされた問題ではなく、大きなシステム（例えば政府や経済）からくる問題に直面していることがあります。我々はシステムを一つの実体（一人の人間、一企業、一経済など）で把握しがちですが、システムはいくつかの実体から構成されていることをたえず認識しておくことが大切です。

### 2.3 システムズ・シンキングとシステム・ダイナミクス

フォレストーによるシステムズ・アプローチには、定性モデルを用いるシステムズ・シンキングと定量モデルを用いるシステム・ダイナミクスがあることは既にお話ししました。そのためのツールとして、システムズ・シンキングでは、因果関係図と時系列挙動図が準備されており、システム・ダイナミクスでは、フロー・ダイアグラムが準備されています。因果関係図とフロー・ダイアグラムとは、Powersim Studio を使って描きます。

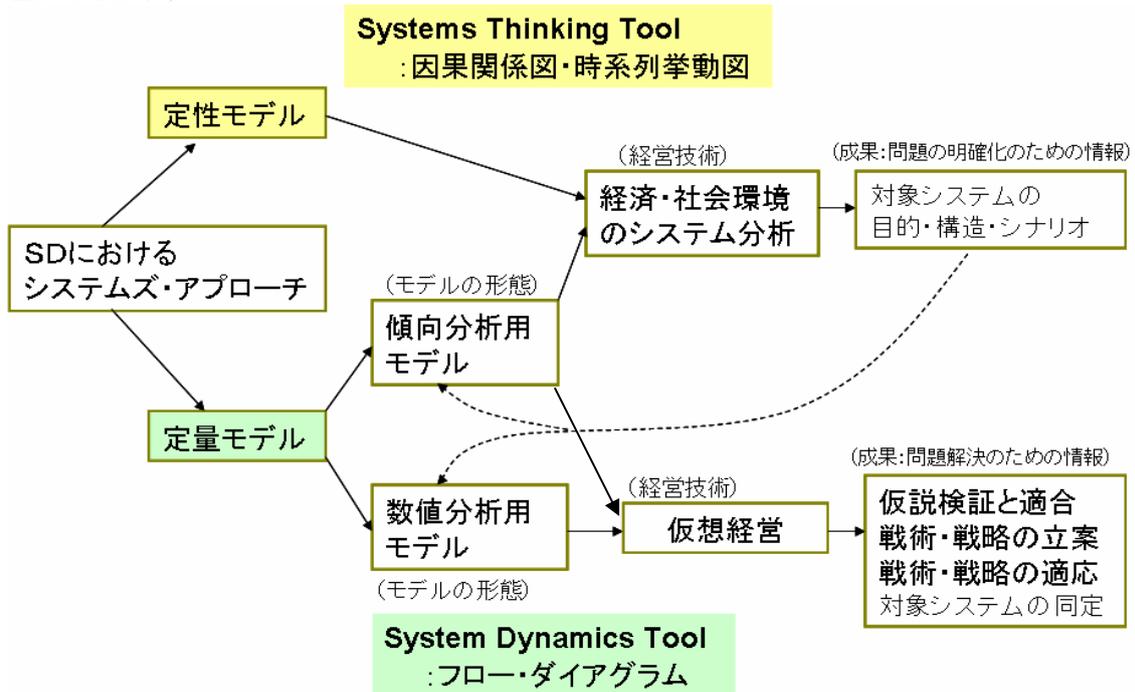
定性モデルも定量モデルも、対象とするシステムの構成要素とその因果関係を図式化したものです。定性モデルでは、その構成要素の時系列の挙動が時系列挙動図として与えられます。定量モデルでは、その構成要素の時系列の挙動は、初期値を与えてその後はその構成要素に定義された方程式により逐次計算されます。定量モデルの構成要素の中には、外部から条件として与える外生変数と、時間経過により内部で計算される内生変数とがあります。

したがって、定性モデルでは、構成要素に対する既存のデータ、あるいは主務者のメンタル・モデルに基づくイメージが、その要素について推測される挙動となります。このようにして推測されたモデルのダイナミックな特性から判断して、問題を解決できるように、モデルの構造を変更すること

になります。

一方、定量モデルでは、外政変数に初期条件を与えた後は、モデルの構成要素に組み込まれた方程式によって挙動が逐次計算されますから、複雑なシステムでは主務者が想定していなかったような特異な挙動が発生する可能性も出てきます。

定性モデルと定量モデルとは、それぞれに期待する機能と目的とがあるわけですが、それを下図に示します。



さて、対象とするシステムに関して、関係者間で現状認識、システムのあるべき価値観、将来の目標などが明確な場合には、そのシステムズ・アプローチをハードシステム・アプローチと呼んでいます。この場合には、システムの構造が明確に決まっていますから、定量モデルとして数式で表現できます。したがって、取り組む問題に対して、例えば今後のあるべき条件と到達できる目標などの解は、数学的な手法で求めることができます。

しかし、経営問題では、関係者間で現状認識、システムのあるべき価値観、将来の目標などについてコンセンサスが得られていない状況が多々あります。このような場合には、直ぐに定量モデルを構築することはできませんから、定性モデルを準備してコミュニケーションを図り、何らかの合意点を見出す作業の段階が必要になります。このように、モデルの仕様が確定していないモデルに対するシステムズ・アプローチのことを、ソフトシステム・アプローチと呼んでいます。この分野では、チェックランドほか(Peter Checkland & Jim Scholes)によるソフト・システム方法論(SSM)が良く知られています。

フォレストーによるシステムズ・アプローチでは、定性モデルとして因果関係図と時系列挙動図とを使うわけですが、これらでは人の頭の中のメンタル・モデルを図式化するために使うこともできません。各関係者のメンタル・モデルをこれらのツールを使って図式化してコミュニケーションの材料と

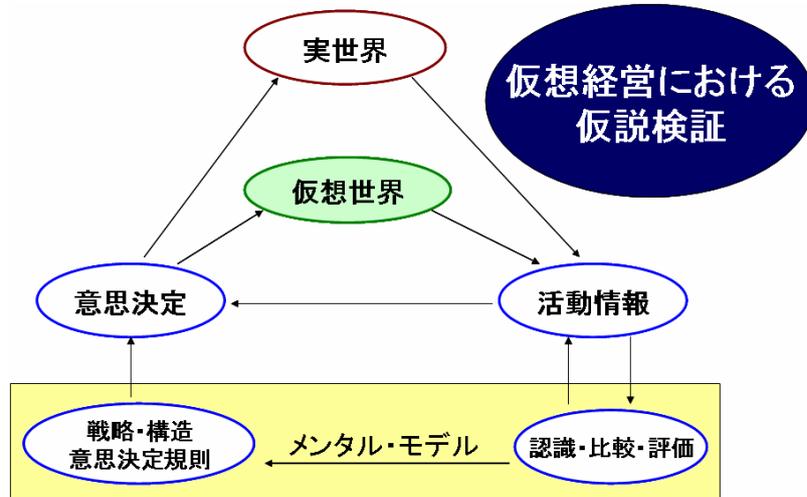
します。完全なコンセンサスを得ることは無理としても、関係者間のアコモデーションに到達することを目指します。その結果として何らかの統一見解が得られるなら、対象としているシステムを定量モデルとして表現することも可能になります。

定量モデルのツールであるフロー・ダイアグラムは、それ自身が因果関係図でもありますから、取り上げている対象が小さくて簡単なシステムであるなら、定性モデルなしで定量モデルを構築して関係者間のコミュニケーションを図ることには何ら問題がありません。しかし、ある程度以上大きなシステムを対象として関係者間でコミュニケーションを図りたい場合には、詳細で緻密なフロー・ダイアグラムでは複雑すぎてコミュニケーションの手段としては使えません。こんな場合には、粗の密度の因果関係図を作り、それをコミュニケーションの材料として供する必要があります。このように、因果関係図はその使用目的に合わせて、その内容の粗密を設定して作成できることが大きな特長です。

前述の図を見て分かるように、定量モデルには、傾向分析用モデルと数値分析用モデルとがあります。数値分析用モデルは結果の数値の精度を数値計算として保証できるモデルです。一方の傾向分析用モデルは、定性モデルにその使い方が似ていて、数値の精度を保証するのではなく、増加する、減少する、変化が大きい、小さいなどのように、操作変数に対して結果変数がどの傾向で変るかを確認するためのモデルです。

実は、公表されているほとんどのモデルはこの類に属していて、有名な“成長の限界”のワールド・モデルも正に傾向分析用モデルです。したがって、出力されている数値そのものにこだわって注目するのではなく、操作変数を変えたときの結果変数の変化の傾向に注目すべきモデルです。また、当然のことですが、傾向分析用モデルと数値分析用モデルとでは、モデルの検証方法が全く異なることに留意する必要があります。

さて、経営問題においては、定性モデルを使って関係者間でアコモデーションに到達したり、一部の人たちが標榜する“気付き”の道具として使ったにしても、企業経営においてそれだけでは問題解決までには繋がりません。少なくとも傾向分析用の定量モデルを構築して、“仮想経営における仮説検証”を行なう必要があります。そのプロセスを以下の図に示します。



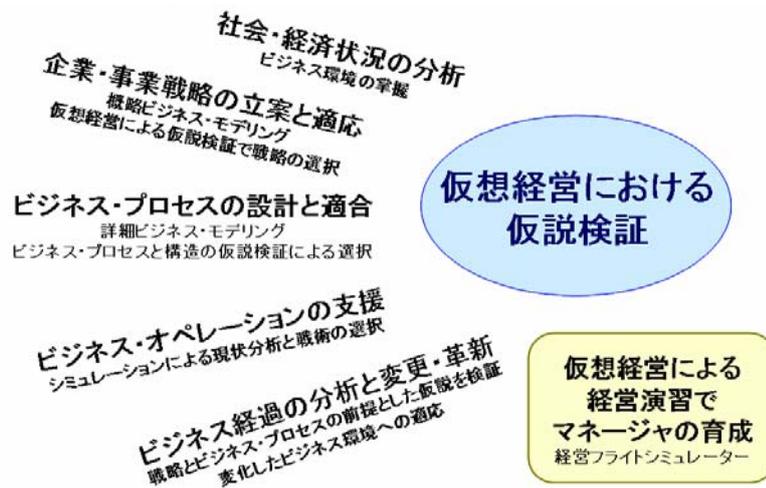
“実世界→活動情報→意思決定→実世界”のトライアングルが、実経営に対するフィードバック・ループです。これだけだと、活動情報に問題を見出して、対策として仮説を考えついたにしても、それを前もって試すことなく、直ちに実適用する以外に対処方法がありません。従来のように環境の変化が緩やかで、経験の延長線上に問題解決の解が存在する時代にはそれでも良かったのですが、昨今のように環境変化が激しく、経験だけによる意思決定ではリスクが大きくなってくと、出たところ勝負の意思決定では企業経営を大変危険な状況に陥らせる可能性が出てきました。

そのために、自社の経営モデルである“仮想世界”を構築して、“仮想世界→活動情報→意思決定→仮想世界”のトライアングルを使って、短サイクルで仮説検証を実行します。その結果得られたリスクの小さい、あるいは結果変数が最適な意思決定を実経営のトライアングルにおける意思決定として実行します。この対応により実経営でのリスクヘッジが可能になります。

メリットはそれだけではありません。経営トライアングルでの意思決定の過程では、経営者・経営スタッフ・管理者のメンタル・モデルを介して、活動情報を認識して意思決定を行っています。上の図で経営トライアングルの下にある四角の枠で示したメンタル・モデルがそれです。ここでの認識・比較・評価から戦略・構造あるいは意思決定規則が、仮想世界のトライアングルが回転するたびにリファインされ、レベルが高度化するのはです。すなわち上記の関係社員のポテンシャルアップが図られることになりまますから、次期からの問題発生に対応する仮説の設定が段々との確に変わっていきます。

このことは、上述のような経営オペレーションを通して実行されますが、それと共にこのモデルを社員研修において業務と切り離して使うことで、経営フライトシミュレーターとしての効果を得ることもできます。これからの企業経営では短時間に大きく変化する環境に対して、いかに迅速にかつ的確に対応できるかが勝負になります。そのためには“できる社員”の育成が何としても必要です。年長者が経験によって育成し難い状況になってきていますから、考えるための方法論であるシステム・ダイナミクスを社員の育成にも大いに活用すべきと考えています。

ここまででシステム・ダイナミクスに基づく経営モデルを、仮説経営において活用することをお話ししました。なお、以下には、その仮想経営における仮説検証が活用できるビジネスのステップについて概要を示しておきます。



### 3. モデリングの基礎

#### 3.1 範囲の設定

システムズ・アプローチのモデルを構築する際には、“モデルの範囲の設定”で行き詰ることが多いようです。システムズ・アプローチでは、問題をモデル化するのであり、システムをモデル化するものではありません。例えば、会社のモデルを作ろうとしてパソコンの前に座ったら、決して会社全体のシステムをモデル化しようとするべきではありません。その考え方は労多くして無駄を生むだけです。

では、どこまででモデル化を止めるべきでしょうか？ 別の見方をすると、システムはあるシステムのサブ・システムですから、どの上下の階層のシステムまでモデル化に含めるべきでしょうか？ 逆に言うと、モデル化にはどこから外は除外すべきでしょうか？

モデルを作る多くの人が経験することですが、システムの要素をどんどん追加していくうちに、だれにも理解することのできないようなモデルになってしまうこととなります。日用品にもありますが、汎用品ほど役立たないものはないのです。

したがって、前述の質問の答は、“一つの問題に限ってモデル化をする”と言うことです。すなわち、一つの問題に対して一つのモデルということです。こう考えると、モデル化において妥当な範囲を決めることができます。取り上げるその一つの問題に関係のない事象はすべて排除するのです。例えば、取り上げている企業問題において人事部が関係なければ、人事部をモデルに含めないのです。

モデル化とは現実世界のほんの一部を表現することに過ぎません。ある問題をモデル化するということは、その問題を正確に表現する部分のみを含めるべきなのです。その結果としてシステムの事象をよく表わすことができれば、そのシステムの仮想経営(シミュレーション)により、いろいろと条件を変えて結果をみることができます。

このようにシステムズ・アプローチでは、モデル化の範囲を設定することが、最初に取り組むべき大事なステップです。

#### 3.2 問題の定義

システムズ・アプローチでは、“問題の定義”が非常に重要です。モデルは現実世界の一つの表現ですから、解決しようとしている問題を正しく表現したものでなければ意味がありません。それを考えるあまり、ある問題を取り上げる場合に、その周辺のあらゆる現実をモデルに含めようとしがちになります。もしそうするなら、できあがってくるモデルは前述のように、だれにも理解することのできないモデルになってしまうことでしょう。問題を絞ることにより、モデル化の範囲を設定するのも楽になり、でき上がったモデルに対して、より大きな確信をもてるようになります。

問題を定義するのに最もよい方法は、その事象を時間の流れに即して描いてみることです。例えば、自分の職場でモデルを構築しようとしていて、“どうして在庫がこんなに変動するんだろう”と感じたなら、変動している在庫のグラフを描いて、そのグラフ上に分かっている在庫量を書き込んでみてください。このように時系列グラフやその上に現実の数字を記載して眺めることにより、問題

がはっきりして、構築するモデルの目的が明確に定義されるはずですが。

### 3.3 フィードバック・ループ

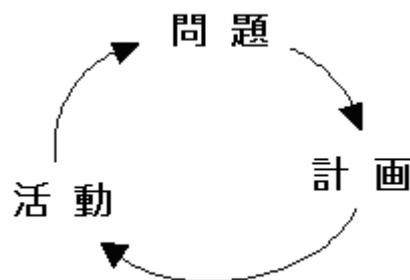
対象とする問題において、原因の情報が入力されシステムの中で加工され、その結果の情報が出力されます。フィードバックとは、その出力された結果の情報が再び原因の情報として入力されることを指しています。そしてその状態のループをフィードバック・ループと呼んでいます。もう少し広い意味ではクローズド・ループと呼ばれています。

一方、世の中では、目の前で発生したイベントに反応するだけと言うような“やりっ放し”が多いものですが、これをオープン・ループと呼んでいます。フィードバックはありません。下図は、オープン・ループの考え方を表現したものです。

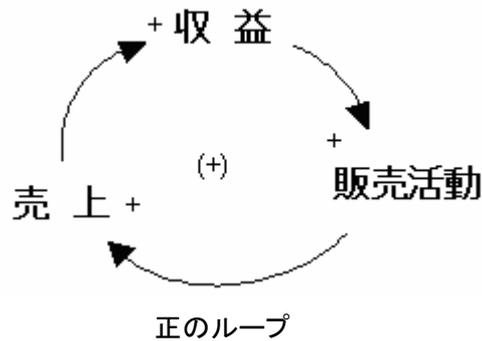
## 問題 → 計画 → 行動

問題が特定され(例えばマーケットシェア)、計画がたてられ(営業部隊の増強など)、行動に移されます(営業マンの採用)。そして問題は解決します。この考え方のようにフィードバックがまったく考慮されていないというのは間違っていることがほとんどです。実際のシステムではアクションは、必ずしも正しい方向へではないのですが、システムを何らかの形で変化させるので、その時点で問題を再定義する必要が出てきます。新しい問題に対しては違った計画が必要ですから、とるべき行動も別の内容になります。このようにほとんどの問題解決の過程はインタラクティブなプロセスなのです。

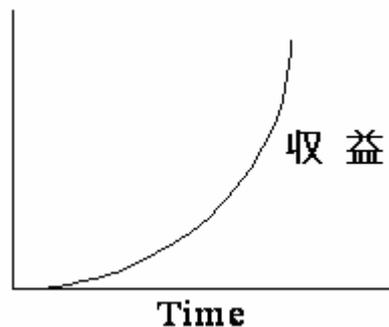
実世界の多くの状況において、オープン・ループからフィードバック・ループへ、考え方を転換する必要があります。フィードバック・ループとは、前述のように原因と結果のサイクルを示す閉じた系のことですから下図のように表現できます。



フィードバック・ループには正のループと負のループがあります。正のループは、“レインフォーシング・ループ”(Reinforcing Loop)と呼ばれ、ループを回るときに現象が増長されていきます。正のループは、バクテリアの増殖から銀行の預金残高の増加にいたるまで、我々の知っているあらゆる成長のプロセスの基になっています。例えば下図に示すように、企業の収入の伸びも正のループによって表現されます。

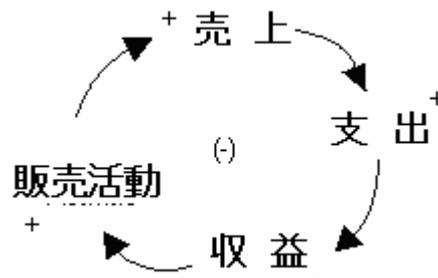


収入が増加すれば、営業活動に割ける資金も増えます。それによって以前より多くの営業マンを雇うことができたり、現在の営業部隊に以前より長い時間の手当を支払うことができます。その結果、営業活動が促進され、より多くの受注を獲得し、さらに企業の収入を増大させることができます。正のループでは、下図で示すような指数増加の挙動になります。



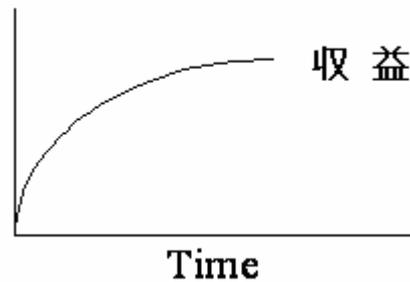
正のループの現象

一方、負のループはゴール追及型のループです。システムをゴールから遠ざけようとする力に対して、それを打ち消すような力が働きます。負のループは、システムを安定化させるのです。例えば企業の収入を題材にした前述の例は、もう1つの要素である“支出”を足してやれば、下図のように負のループの説明にも使えます。



収入の限界を表す負のループ

このループでは、受注が増えれば収入も増えますがそれと共に、販売量に伴う運転費などのコストも増えますから支出は増加します。支出が増えた結果、収益は減ることになります。このシナリオでは受注の伸びが、全部ではないとしても、支出の増加によってキャンセルされてしまいますから、負のループは下図のような頭打ちの挙動を引き起こすことになります。



負のループの現象

この負のループはシステムをある状態に収束させることから、“バランシング・ループ” (Balancing Loop)と呼ばれています。

実際のシステムでは、独立した正のループだけとか、負のループだけを持つということはほとんどありません。この2つのタイプのループは、複雑なシステムの中で互いに干渉し合います。正のループは増加や減少をもたらす、負のループは安定、目標への収斂、ある値の周りでの振動といったものをもたらします。

システムの中でどの要素がどの要素に影響しているのか、またその作用は正か負かという“システムの構造”の複雑さを解きほぐすには、フィードバックのプロセスを理解することがその第一歩となります。しかしだからと言って、因果関係図で表現したフィードバック・ループだけで、システムの構造を完全に表現したり、システムの挙動を完全に理解することはできないのです。ただ、注意しておかないといけないのは、システムズ・シンキングの至上主義者の中には、フィードバック・ループを用いた因果関係図を駆使することで、システムの構造を理解して、ほとんどのことを学ぶことができると考えている人たちもいることです。

一方、定量モデルを使うシステム・ダイナミクスを方法論として活用している人たちは、ストックやフローの概念と、それがフィードバック・ループとどう関係しているかについても重きを置いています。したがって、システム・ダイナミクスに取り組んでいる人たちは、経営問題を解決するためには、因果関係図だけではなく少なくとも傾向分析用の定量モデルを使ったシステム・ダイナミクスによるアプローチが不可欠であると考えています。

### 3.4 レベルとフロー

システム・ダイナミクスで用いる定量モデルには、レベル、フロー、補助変数、定数、リンク線などが含まれていますが、中でもレベルとフローが重要な要素です。“レベル”はストックとも呼ばれていますが、“溜まるもの”(蓄積)です。“フロー”は“溜めるもの”(流入)または“減らすもの”(流出)です。定量モデルの中では、フローだけがレベルの蓄積量を変化させることができます。

#### (1)レベル

我々の身の回りではいろいろなものが溜まります。掃除をすると同時にほこりは積もり始めるし、ものを食べれば胃袋が満たされます。事故で道路が遮断されると車が道路にあふれてきます。在庫というのも企業における大事な溜まるものです。システム・ダイナミクスではこういった“溜まるもの”のすべてをレベルと呼んでいます。

レベルは現実世界ではステータスを意味していますから、その値はある特定の時点におけるシステムの状態を表わしていることとなります。時刻を停止すると、レベルもある一つの状態で停止し、観察や測定が可能となります。定量モデルの中でレベルは動詞ではなく、名詞で表現されます。ある意味ではレベルは静的で動きのないものと言えます。

レベルには顕著な特徴があります。レベルは溜めるものや減らすもののフローによってのみ、その値を変化させますが、フローによる変化には時間がかかります。この変化に必要な時間のことを、後述しますが“遅延”(Delay)と呼んでいます。どんなに小さくとも必ず遅延は生じています。レベルが名詞だとすれば、フローは動詞(動名詞)と表現することができるでしょう。フローは“溜めるもの”と“減らすもの”がありますが、これらはレベルに作用してレベルの値を変化させます。したがって、フローがなければレベルは決して変化しないし、ダイナミックな現象も発生しません。フローにはその時間当たりの流量を設定するためのレート変数あるいは補助変数が接続します。

### 3.6 遅延

“遅延”はシステムの要素では見落とされやすいものです。多くのシステムにおいて挙動の不安定性の原因となっていて、問題解決をより難しくしています。端的に言うと、遅延によりシステムに問題が発生していたとしても、それにより挙動に異変が現れて問題に気づくまでに、長時間が経過してしまうことが多いのです。例えば、20年もの間、システムに潜在してきた問題が発見されたとしても、それを解決するのは、先週突発的に発生した問題の解決に比べて、はるかに難しいのです。

経営問題で例えると、経営者が日常やっている昔ながらの場当たり主義的な問題解決では隠れていた、長い間に忍び込んだ長期的な問題が、手に負えないくらいになってから表面化することなどは、会社が消滅する原因にもなるのです。

問題の発生と問題の表面化の間の遅延が大きいほど、その問題の解決は難しくなります。遅延が発見されると、問題を解決しようとする行動にも影響が及びます。長い間に大きくなった問題は一夜にして解決するわけがないのに、我々は速効性のある解決を求めようとします。たとえば、国家予算を正常化するために累積している赤字国債を一掃する即効薬を求める心情などは正に上述の指摘に相当します。回復への道のりは長いものだという警告に耳をかさない傾向があるのも、遅延のせいであることが多いのです。

#### 4. Powersim Studio の操作の初歩

Powersim Studio は実際のビジネスの世界に適用することを前提として作成されていますので、その機能は年々充実しています。したがって、この小冊子で全てを解説することはできませんから、ここでは Powersim Studio を初めてインストールした人が、操作を開始できるきっかけがつかめる程度の初歩的な情報に絞ってお話することにします。

使い始めるきっかけがつかめた皆さんは、Powersim Studio の Help 機能に収録されている詳細な電子マニュアルを活用したり、あるいは POSY 社のオリエンテーション・コースを受講するなりして、各自で中・高度のレベルのモデリング技術を修得してください。

cf. POSY 社のオリエンテーションコース : <http://www.posy.co.jp/training-f.htm>

##### 4. 1 Powersim Studio の種類

使用目的に合わせて、Powersim Studio には複数の製品が準備されています。そのうちの Studio Demo 版と Studio Express 版は、ツールの機能を評価していただくために Powersim 社が無償で提供している評価版です。これらの評価版をダウンロードするには、付録 2 : Ps Studio 8 Demo/Express の内容に沿って、Powersim 社のダウンロードのページに進んでください。

##### Studio Demo 版

最上位の製品版である Studio Enterprise と同じ機能です。

有効期間 : ダウンロード後 30 日間

再利用 : 同じ PC 上にはインストールできません。

POSY 社に連絡すると、1 回延長 (2 週間) ができます。

##### Studio Express 版

分析向けの商品版である Studio Professional と同じ機能です。

有効期間 : ダウンロード後 60 日間

再利用 : 同じ PC 上で何度でもインストールして使用できます。

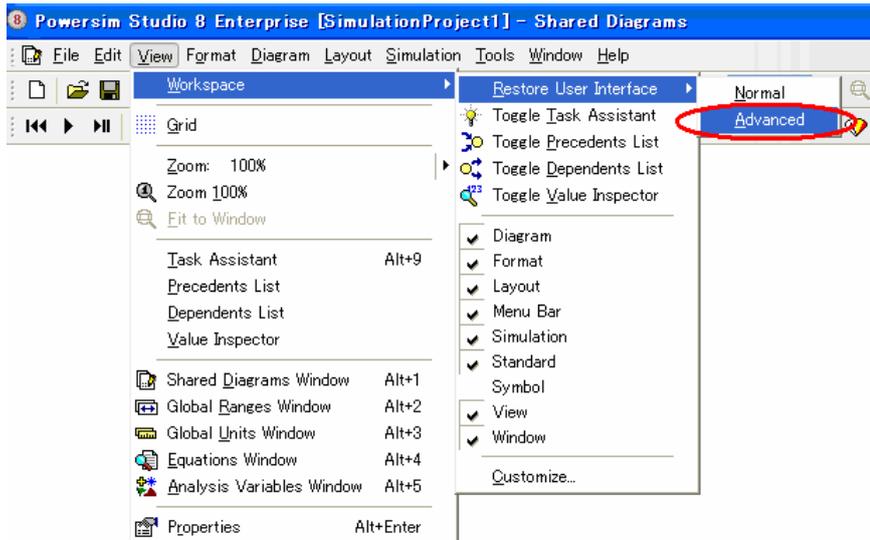
各製品の概略の機能を下表に示します。

製品名 Studio-	シミュレーション	モデリング	Excel接続	最適化 リスク評価	ネットワーク接続 外部組込 ランタイム組込
Enterprise	●	●	●	●	●
Demo	●	●	●	●	●
Expert	●	●	●	●	
Professional	●	●	●		
Express	●	●	●		
Academic	●	●	●	●	
Executive	●		●	●	●
Cockpit	●		●		

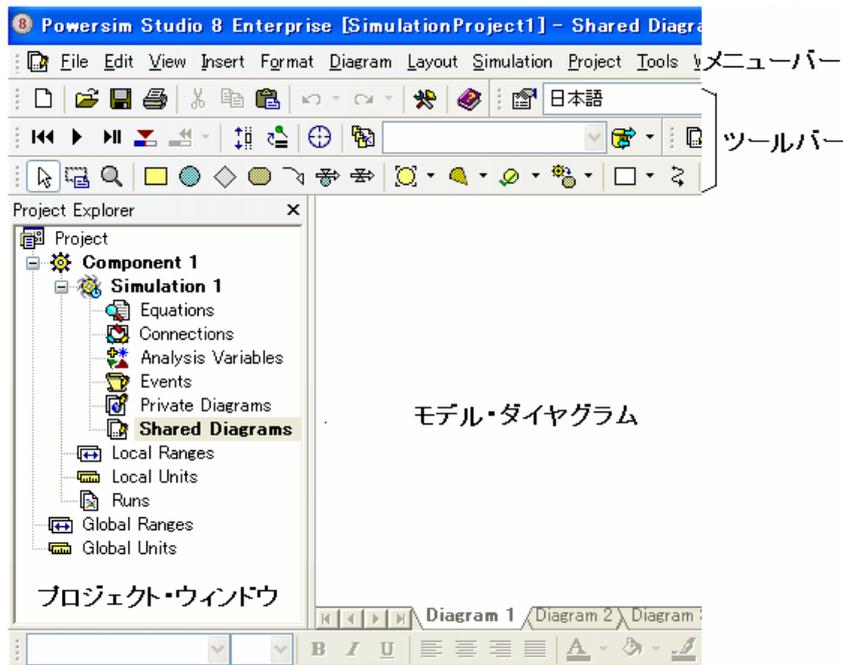
## 4. 2 最初の画面

Powersim Studio には、二種類のトップ画面があります。Normal 画面と Advanced 画面です。全くインストールしたことがない PC 上に初めてインストールすると、Normal 画面が立ち上がります。しかし、機能が充実した画面は、Advanced 画面なので使用開始時にそちらに切り替えてください。一度切り替えるとその後はその状態が保持されます。

最初に立ち上がった Normal 画面上の最上段にあるメニューバーの“View”から、“Advanced”を選択している様子を下図に示します。



Advanced 画面が開いた状態を以下に示します。Powersim Studio の画面構成と操作性とは Microsoft Windows に倣っていますので、多くの人に馴染み易いと思います。



画面の上端にツールバーとメニューバーがあります。その下の画面は左右に分かれていて、左側をプロジェクト・ウィンドウ、右側をモデル・ダイアグラムと呼びます。左のプロジェクト・ウィンドウにはこのモデルの構成を表示し、右のモデル・ダイアグラムには、モデルとその入出力のオブジェクト(フロー・ダイアグラム)を表示します。

モデルの構造を表示しているプロジェクト・ウィンドウから説明します。Powersim Studio のモデルのファイルの拡張子は“sip”です。このファイルの全体を“Project”と呼んでいます。この Project の中には、複数のモデルを含むことができます。そのモデルのことを“Component”と呼びます。モデルとは単独でシミュレーション計算を実施できる機能を備えているまとまりを指しています。Project に Component を追加するには、プロジェクト・ウィンドウの最上端の Project 名を右クリックして表示されるプルダウン・メニューの中の“Add Component”をクリックします。

さらに、各モデルではシミュレーション条件の異なる計算を実施するために、計算条件の組み合わせを保持したシミュレーション・ケースを設定して保管しています。それには2種類あり、一つは通常の確定的シミュレーションを行なう“Simulation”で、もう一方は確率的シミュレーションを行なう“Risk Analysis”です。これらはそれぞれ複数個設置することができます。同じ Component におけるシミュレーション・ケースでは、モデル本体の構造は全て同じです。

前者の“Simulation”において設定できる計算条件は、入出力条件、表示条件、期間や刻み時間幅などです。後者の“Risk Analysis”では、確率的シミュレーションとして、モンテカルロ法またはラテン・ハイパーキューブ法を選択できます。

あるモデル(Component)に Simulation または Risk Analysis を追加するには、プロジェクト・ウィンドウのモデル名の行を右クリックして、表示されたプルダウン・メニューの中の“Add Simulation”、あるいは“Add Risk Analysis”をクリックして選択します。

Powersim Studio では、サブモデルを組み合わせる大きなモデルを構築したり、あるいは Excel や企業の統合データベースである SAP BI、Oracle あるいは Access と接続したモデルを構築できます。それらの構築も、プロジェクト・ウィンドウから操作して行ないます。

モデル・ダイアグラムには、共通画面(Shared Diagram)と個別画面(Private Diagram)とがあります。それらはプロジェクト・ウィンドウ上で切り替えます。共通画面には、モデルの本体部分と各種のシミュレーション・ケースに共通な入出力部分を描きます。個別画面には、それぞれのシミュレーション・ケースでのみ使われる特殊な入出力部分を描きます。

#### 4.3 メニューバーとツールバーでよく使うアイコン

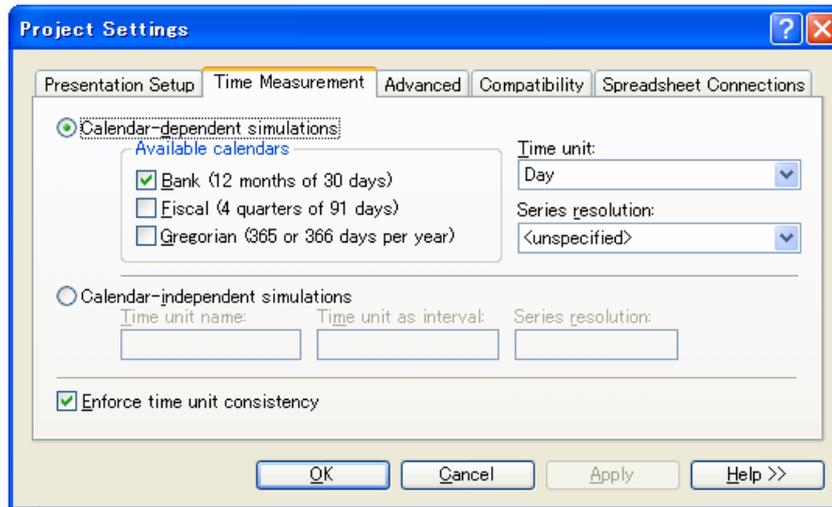
多数の機能がありますが、その中で頻繁に使うものだけについて、以下で簡単に説明します。

##### (1) メニューバー



先にも説明しましたように、Powersim Studio は、Microsoft Windows に倣っていますので、上に示したメニューバーの機能の使い方については、使用者が類推し易いと思います。しかし、“Simulation”と“Project”とは独特なメニュー項目ですから以下で内容を説明します。

メニューの“Project”のプルダウン・メニューの中の“Project Setting”で、プロジェクト全体の条件を設定します。その中でも、“Time Measurement”タブが重要ですから、以下にその図を示します。



カレンダーとして、3種類の設定ができます。Bank 暦、Fiscal 暦、グレゴリアン暦です。

Bank 暦は、30 日／月 × 12 月 = 360 日／年

Fiscal 暦は、91 日／四半期 × 4 四半期 = 364 日／年

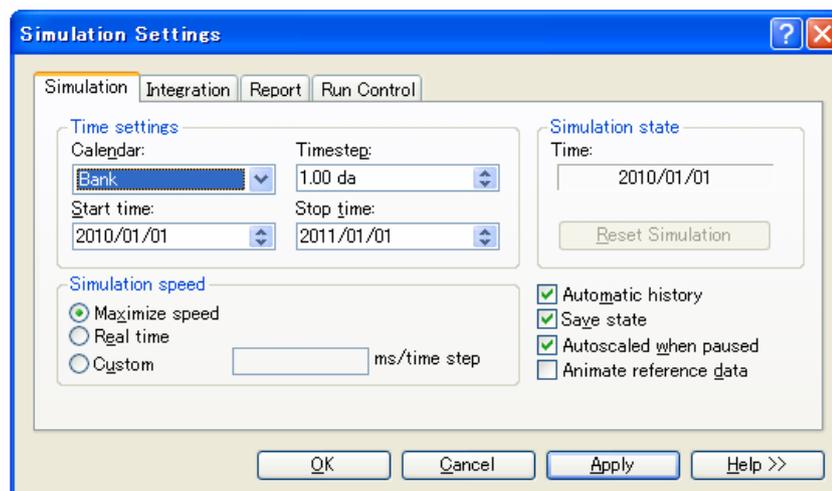
グレゴリアン暦は、通常の暦です。

ここではカレンダーの候補を設定し、実際にシミュレーションで使用するカレンダーは、Simulation Setting で選択します。

“Time unit”は、シミュレーションで使用される時間単位で、デフォルト値は“日”です。

“Calendar-independent simulation”は、カレンダーを使用しないシミュレーションの場合に選択し、引き続きその条件を設定します。

メニュー“Simulation”のプルダウン・メニューの中の“Simulation Setting”で、プロジェクト・ウィンドウの中のシミュレーション・ケースごとのシミュレーション条件を設定します。Simulation Setting の中の“Simulation”タブを以下に示します。



Calendar: メニューの“Project”で選択したカレンダーの候補の中からシミュレーションで採用するカレンダーの種類を選択します。

Timestep: シミュレーションの刻み時間

Start time: シミュレーションの開始時刻 (1日未満の表現 HH:MM:SS)

Stop time: シミュレーションの終了時刻 (1日未満の表現 HH:MM:SS)

Simulation Speed: ミュレーションの表示スピードです。

シミュレーションの途中経過を示しながら関係者間でコミュニケーションする場合には、シミュレーションの途中経過を理解し易いように、ゆっくりした表示スピードに設定するのが効果的です。

## (2) ツールバー

ツールバーは、形式をカスタマイズできますが、以下にデフォルトの形式を示します。

2行目のシミュレーション・コントロール部、3行目のモデルのシンボル部と入出力表示部について説明します。



## シミュレーション・コントロール



リセット・ボタン

パーマネント設定していない変数は、モデルの属性枠で設定した初期値に戻る

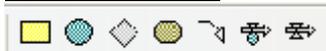


実行ボタン



1プッシュで1刻み時間(timestep)のシミュレーションが進むボタン

## モデルのシンボル



レベルのシンボル



補助変数のシンボル



定数のシンボル



サブモデルを保管する箱のシンボル

以上の4つのシンボルをダイアグラムへ配置する方法

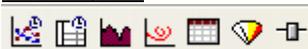
- ① ツールバー上の相当するシンボルのアイコンをカーソルで左クリック
- ② カーソルにシンボルが付き、その左上に配置位置のターゲットのクロスが表示
- ③ クロスの中心がダイアグラム上の所定の位置に来たら左クリック
- ④ シンボルが選択された状態で、任意な言語で名前を入力

-  情報リンク線: シンボル間を接続すると、矢元から矢先に瞬時に情報が伝達
- ① ツールバー上の相当するシンボルのアイコンをカーソルで左クリック
  - ② カーソルにシンボルが付き、その左上に配置位置のターゲットのクロスが表示
  - ③ クロスの中心が矢元のシンボルの中に入ったら左ボタンをプッシュ(オン)
  - ④ プッシュしたままドラッグし、クロスの中心が矢先のシンボルに入ったらオフ
- レベルや定数に情報リンク線を張ると実線が点線に変わるのは、初期値としてシミュレーションの開始時にのみ情報伝達が行われることを意味します。
- 情報伝達が実行されても、矢元の情報が伝達されて無くなることはありません。

-  レート付きフロー: 矢元から矢先に搬送されるため、矢元のものは無くなり、その量だけ矢先に移ります。
- 丸はレート(rate)と呼ばれますが、前述の補助変数がリンク線でフローのバルブ(向かい合う三角形)に接続されたものと同じです。
- 矢元の雲の絵は無限のソース(source)を表し、矢先の雲の絵は無限のシンク(sink)を表しています。

-  フロー
- 上記二つのフローをダイアグラム上に配置する方法
- ① ツールバー上の相当するシンボルのアイコンをカーソルで左クリック
  - ② カーソルにシンボルが付き、その左上に配置位置のターゲットのクロスが表示
  - ③ クロスの中心が矢元のレベルの中、または所定の位置に来たら、左ボタンをプッシュ(オン)
  - ④ プッシュしたままドラッグし、クロスの中心が矢先のレベルの中、または所定の位置に来たら、左ボタンをオフ

## 入出力表示



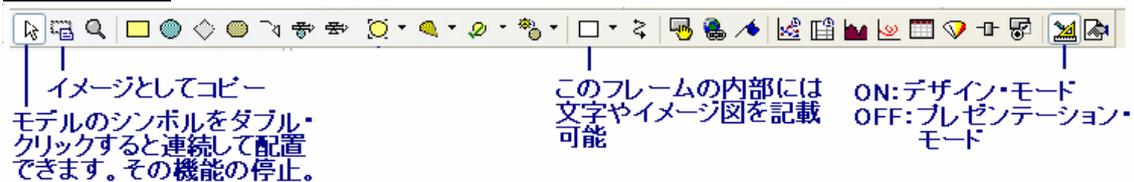
-  時系列グラフ
-  時系列テーブル
-  横軸が配列要素のグラフ
-  位相面グラフ
-  その時点のデータを表示するテーブル
-  扇形インディケータ
-  スライドバーもしくは棒形インディケータ

以上の入出力用のシンボルをダイアグラム上に配置する方法

- ① ツールバー上の相当するシンボルのアイコンをカーソルで左クリック

- ②カーソルにシンボルが付き、その左上にシンボルの“左上”の配置位置のターゲットのクロスが表示
- ③クロスが中心がダイアグラム上の所定の位置に来たら左ボタンをプッシュ(オン)
- ④プッシュしたまま右下にドラッグし、所定の大きさになったらオフ

### その他の機能



 **イメージとしてコピー:**このアイコンで囲った領域がイメージとしてコピーされ、ワードなどの任意のアプリケーションに貼りつけることができます。

- ①ツールバー上の相当するシンボルのアイコンをカーソルで左クリック
- ②カーソルにシンボルが付き、その左上にシンボルの“左上”の配置位置のターゲットのクロスが表示
- ③クロスが中心がダイアグラム上の所定の位置に来たら左ボタンをプッシュ(オン)
- ④プッシュしたままコピーしたい部分を右下に向かってドラッグし、所定の範囲を覆ったらオフ
- ⑤予定したアプリケーション上でペースト(貼りつけ)

 **フレーム:**この枠の中には文字とイメージ図を表示できます。

#### 4. 4 Help 機能の使い方

Powersim Studio には、紙に印刷された詳細なマニュアルはありません。メニューバーの“Help”をクリックして開くプルダウン・メニューの中の“Contents”を選択すると右に記す Help のページが開きます。

ここでは目次のタブが表示されていますが、いわばこれが詳細な電子マニュアルの目次で、バージョンアップの度に更新されています。

通常は、“キーワード・タブ”か、“検索タブ”を開いて、探したい項目を英文で記入します。それにマッチした内容がこの画面の右側に表示されます。

なお、この目次の Help の内容を PDF ファイルとして廉価に販売しています。PDF ファイルとして保管された詳細な電子マニュアルを希望される方は、

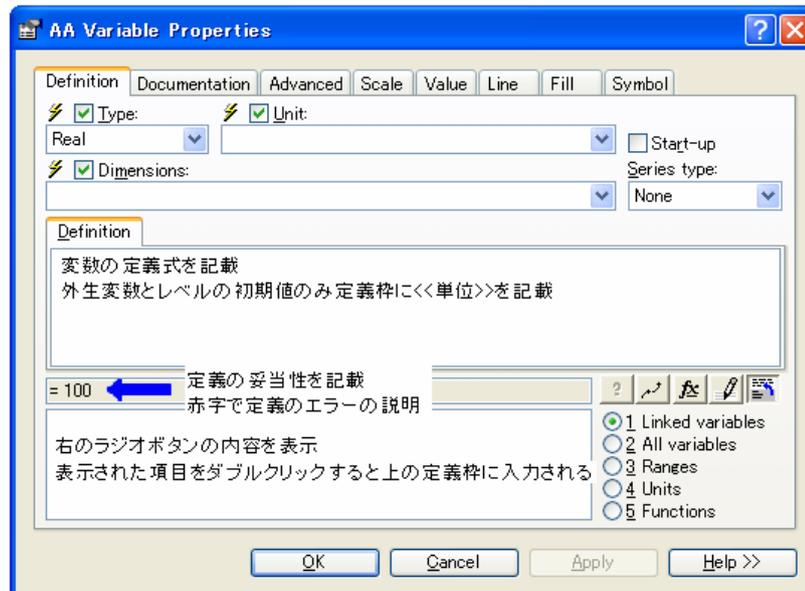


POSY 社までお申し込みください。

#### 4.5 変数のシンボルと入出力表示部の属性の定義

##### (1) 変数のシンボルの属性

およそ1枚のダイアグラム分のモデルの構造を描いたら、モデルの周辺部から中心に向かってそれぞれの変数の方程式を定義します。その定義は、各変数のシンボルをダブル・クリックすると開く下図の属性画面で行ないます。



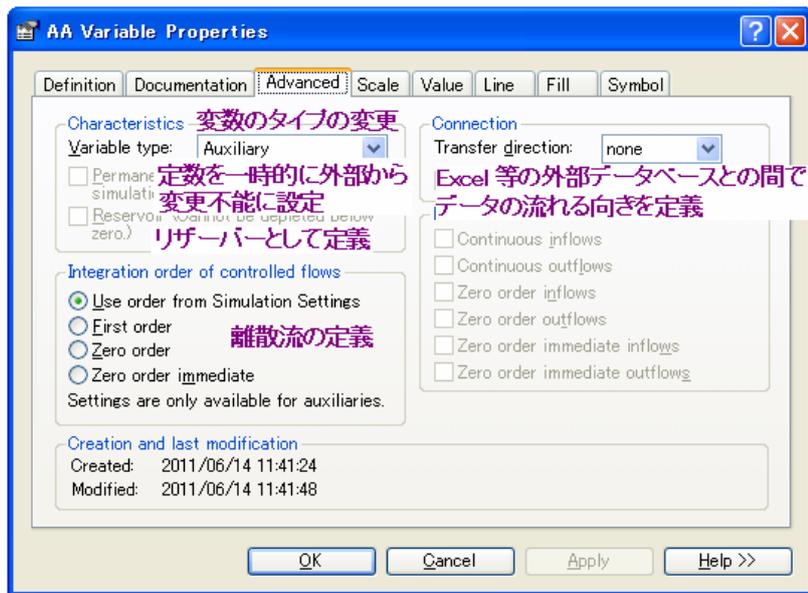
方程式は定義枠 (Definition) に記入します。この変数にリンク線で入力されている変数名が、Linked variables を選択することで、定義枠の下の枠に表示されます。ここに表示された変数の全てを定義枠で組み込む必要があります。

下の枠の変数名をダブル・クリックすると、上の定義枠に表示することができます。定義枠の中で単位を意識的に入力するのは、原則的には、定数などの外生変数とレベルの初期値だけです。その他の変数の単位は、Studio の中で自動的に導かれて決められます。例えば、<<da>>のように、“<<”と“>>”とで囲まれた文字は単位であることを意味します。

定義枠と下の枠との間の狭い帯状の場所には、定義された結果が黒い文字で表示されます。しかし、定義部分にエラーが残っている場合には、この部分に赤い文字でエラーの内容が表示されます。その内容を読んで定義枠の記述内容を修正してください。

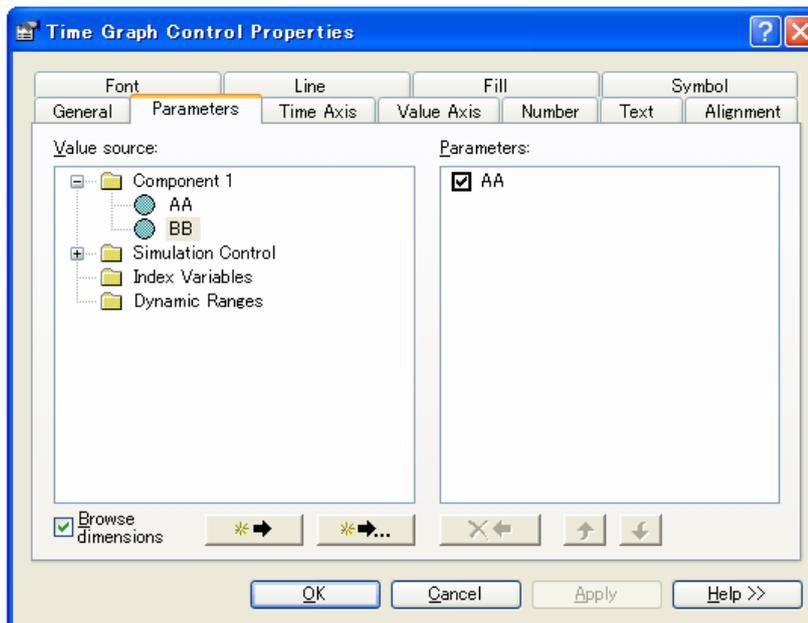
この属性ウィンドウに記載された内容について不明な点が生じた場合には、このウィンドウの右下にある“Help”をクリックして下さい。記載されている単語単位で説明が表示されます。

一般的な定義は、上記の“Definition”タブで行ないますが、少し高度な設定は、以下に示す“Advanced”タブで開く画面上で行ないます。



## (2) 入出力表示部の属性

Studio には7種類の入出力表示部が準備されています。ツールバーに並んでいるそれらのアイコンをダイアグラム上に表示して、その画面をダブル・クリックすると下に示す属性画面が開きます。



入出力表示部には表示する変数名を定義する必要があります。その方法として2種類の方法があります。

- ・ ダイアグラム上で定義する変数のシンボルを、左ボタンを押さえて選定し、そのまま入出力表示部の中まで引っ張ってきて、そこで左ボタンを離します。
- ・ 属性画面の“Parameters”タブで、定義する変数を選定し、下の右向きの2つの矢印の一方をクリックすると、右側の Parameters 枠に表示されます。

左右の矢印の選択についてですが、シミュレーション結果をそのまま表示する場合には左側の矢印をクリックします。一方、参照データとして表示する場合、あるいは一定の間隔ごとの平均とか累積値などの加工したデータを表示する場合には、右側の矢印をクリックして、開いたウィンドウで表示条件を設定します。

#### 4.6 単位の設定

Studio で単位はモデルの構造の論理チェックに使われますから、大変に重要です。単位を全く付けないでモデルを構築することができないわけではありませんが、論理間違いを起こす原因になりますから、正統的に単位を付けることを強く勧めます。

さて、単位は3種類のグループが、国際(SI)単位系で組み込まれています。

##### ①最初から組み込まれている単位

時間単位が中心で、英単語を短縮してStudioの中で変数として定義されています。このグループの単位は、プロジェクト・ウィンドウの Global Units または Local Units を開くと、初めから表示されています。ここに表示されている単位は、変数の単位としてそのまま使用することができます。  
年:yr 四半期:qtr 月:mo 週:wk 日:da 時間:hr 分:min 秒:s

Name	Definition	Plural Name	Documentation
%	0.01	%	Percent
da	24hr	da	Day
deg	(3.14159265358979323846/180)rad	deg	Degrees - Plane angle
grad	(3.14159265358979323846/200)rad	grad	Gradians - Plane angle
hr	60min	hr	Hour
min	60s	min	Minute
mo	30da	mo	Month
period	_TIME	periods	Project Time Unit
qtr	90da	qtr	Quarter
rad	_RADIAN	rad	Radians - Plane angle
s	_SECOND	s	Second
wk	7da	wk	Week
yr	360da	yr	Year

##### ②組み込まれている標準単位と貨幣の単位

Global Units または Local Units には表示されていないので、直ぐには使えませんが、単位系として整合性を保って準備されている2種類のグループがあります。標準単位と貨幣単位です。これらを Global Units または Local Units に表示させるには、前述の①の単位群の下の空白部分で、右クリックして“Add Unit”を選択します。

開いたウィンドウで、Select a build-in standard unit または Select a build-in currency unit を選択してください。そのグループに含まれている単位群を以下に示します。

## 組み込まれている標準単位

m - Meter - Length
kg - Kilogram - Mass
sr - Steradians - Solid angle
mol - Mole - Amount of substance
cd - Candela - Luminous intensity
A - Ampere - Electric current
K - Kelvin - Temperature
km - Kilometer - Length
cm - Centimeter - Length
l - Liter - Volume
N - Newton - Force
J - Joule - Energy
W - Watt - Power
C - Celsius - Temperature
F - Fahrenheit - Temperature

## 組み込まれている貨幣単位

EUR - Euro
BEF - Belgian Franc
DEM - Deutsche Mark
GRD - Greek Drachma
ESP - Spanish Peseta
FRF - French Franc
IEP - Irish Pound
ITL - Italian Lira
LUF - Luxembourg Franc
NLG - Dutch Guilder
ATS - Austrian Schilling
PTE - Portuguese Escudo
FIM - Finnish Markka
SIT - Slovenian Tolar
CYP - Cypriot Pound
MTL - Maltese Lira
AUD - Australian Dollars
CAD - Canadian Dollars
NZD - New Zealand Dollars
USD - US Dollars
GBP - British Pound
JPY - Japanese Yen
KRW - South Korean Won
CHF - Swiss Franc
NOK - Norwegian Kroner
loc - Currency - Locale currency unit

## ③組み込まれていない単位の定義

Studio では以下に記すプロセスで、任意な単位を組み込んで使うことができます。

- 1) 最初から表示されている①の単位群の下の空白部分で、右クリックして“Add Unit”を選択
- 2) Add Unit のウィンドウで、“Enter new name”に追加する新しい名前を入力  
(後の変数の定義作業で煩雑さを避けるために、必ずシングル・バイトで名前を付けること)
- 3) Next ボタンで開いたページの“Category”において、原子単位 (Atomic) あるいは生成単位 (Derived) を選択。(長さを例にすると、“m”は原子単位で、“km”や“cm”は生成単位です)
- 4) Finish ボタンで最初の画面に戻り、単位の設定が正常であることを確認します。  
必要なら、その行をダブル・クリックしてその属性画面を開き、Document タブに日本語の説明を追記します。

## 5. モデルの構築

4章では Powersim Studio の操作の初歩を解説しました。しかし、現在の Studio の機能は充実していて、とてもこのような小冊子で解説できるほど単純ではありません。今後、このテキストの内容を補充したにしても、十分なものにはならないでしょう。使用する皆さんが、POSY 社のオリエンテーション・コースを受講するなり、Studio の Help 機能を活用するなりして、各自の方法で習得していただきたいと思います。

cf. POSY 社のオリエンテーションコース : <http://www.posy.co.jp/training-f.htm>

この5章ではモデリングのプロセスを順を追って説明します。学習の過程で、シミュレーション・モデルの構築はインタラクティブなプロセスであることを忘れないでください。システム・ダイナミック

スに基づくモデリングにおいては、直線的にモデルを作り上げるということはまずありえなくて、一歩進んではまた下がり、それまでやったところを評価してリファインするというようなやり方になります。

正に、これがモデリングの技術であり、これは時には恣意的であるとさえ感じるプロセスです。最終段階になってもそのモデルが正しいとか終わったとは言えないかもしれません。一般的に言えることですが、モデルができあがってからそれを操作する段階より、モデルを作り上げる過程における方が、はるかに多くのことを学ぶことができることだけは確かです。

### 5.1 問題の定義

モデリングは問題の定義から始まります。モデリングしようとする問題の定義こそが、モデル化の全過程における最も重要なマイルストーンです。一見簡単なことのように思われるかもしれませんが、問題となる現象を曖昧な概念で定義しただけでは不十分です。問題を定義するということは、モデルの目的を定義することでもあるのです。モデルは何らかの問題を理解し解決するために作られるものです。また問題はできるだけ正確に定義する必要があります。この定義が、その後の作業すべてのベースとなり、モデリングの範囲と価値を決定するよりどころとなります。狭い範囲に焦点をあてればあてるほど、構造が複雑になりすぎるのを防ぐことができることは言うまでもありません。

数値情報がここでの手助けとなります。在庫の変動を表現するのに実際の在庫量のデータを使用するというように、問題の定義に数字を使うことができれば、そのほうが良いのです。たとえ実際のデータではなくとも、問題を時間軸において考えるためには非常に役に立ちます。需要の季節変動が在庫に対して影響するように、問題が変数間の相互作用に関するもの場合は、変数どうしの関係を図解する必要があります。これにより、どの変数がどの変数に関わっているのかを明らかにするのです。

### 5.2 モデルの範囲の決定

システム・ダイナミックスのモデルは個々の単独変数の挙動ではなく、変数どうしの相互作用の結果としてのシステム全体の挙動をターゲットとしています。問題が正確に定義されると、モデルの範囲を分析することができます。

内生変数とはモデルの中で時々刻々の値が生成される変数です。外生変数とは基本的にはパラメータであり、その値は定数やレベルの初期値のように一般には外部から与えられます。

問題を生み出すのに関係のある変数だけをモデルに含むようにすべきなのです。排除してもモデルの動きに大きな違いを与えないものは排除すべきです。逆に外部のパラメータの変化によってモデルの動きが大きく変わるとしたら、そのモデルは自己完結的ではありません。つまり、外部に影響されすぎているということになりますから、そういう場合にはモデルの範囲を拡張してその外部のパラメータを外生変数としてを取り込むようにしなければなりません。

### 5.3 時系列挙動図ほかデータの収集

取り上げている問題自身に対しては既に時系列挙動図を準備していますが、次のステップとして主要な変数に対しても時系列挙動図を準備します。作ったモデルが自分で考えていたような動きをしない時に、これらの時系列挙動図と比較して、モデルに何が欠けているのか、どこが間違っているのかを分析します。

時系列挙動図や関連するデータを探索するにはいくつかの方法があります。インターネット・ウェブ検索や図書館で数時間を過ごしたり、あるいはそのシステムを良く知っている人にインタビューすることになるかもしれません。自分がそのシステムを一番よく知っている場合は、自分の頭の中のメンタルモデルに基づいて、推測できる挙動をスケッチするだけで済む場合もあります。

兎も角、データを集める方法はどうか、要は自分の問題の範囲を理解することです。したがって、同じ問題にいろいろな人がどのようにアプローチをしたかを調べるのも参考になるだろうと思います。また、同じ問題に対して別のモデル化技法を使ったものの結果、あるいはその長所と短所とから得るものがあることでしょう。

### 5.4 要素の抽出

十分なデータが集まったら、次はモデルのキーとなる概念や要素を抽出する段階に進みます。ここでは問題の定義で述べたモデリングの目的を思い出し、構造化しながら要素を抽出します。この時点では、抽出された要素にフィルターをかけて、それらの要素を無理やり排除する必要はありません。集めたすべての要素に対して整合性のある変数名を付けます。そして、各変数の単位を考えます。

例えば、在庫変数の単位は“個”(ko)や“個数”(kosu)などで表されます。この変数のリストは、この時点でブレン・ストーミング的に生成したもので、今後作るモデルに組み込む変数は、このリストの中から抜き出します。別の言い方をすると、モデルの変数リストは始めに作った変数リストのサブ・セットだと言えます。

モデルの変数リストを作るには、モデリングの目的とモデルの範囲をもう一度確認して、変数リストを狭めていきます。モデリングの目的に関係しない変数はここで捨ててしまいます。どちらかわからないものは、後のモデルの設計工程で役に立つかもしれないので、とりあえずは脇に保管しておきます。

### 5.5 概念モデルの設計

問題の定義と範囲が明確になり、主要変数に関する時系列挙動図のデータも集まり、システムの変数のリストができたら、次は概念モデルの設計に進みます。この段階はモデルの構造である因果関係図とフロー・ダイアグラムの構造を作る段階です。

様々な構築方法がありますが、代表的な二つの方法を以下に記します。

- (1) 先ずモデルの変数リストからそれぞれの2要素間の因果関係を結び付けて、次にそれを組み合わせる因果関係図を描きます。最後に、それを基にフロー・ダイアグラムの構造を作りま

す。

(2) モデルの中心になる部分を選定し、さらにその中で最も特徴的な“レベル”を最初に配置して、そこからフロー・ダイアグラムの構造を作り始めます。因果関係図については、必要があれば、フロー・ダイアグラムを基にして必要な粗密度を考慮した上で新しく描きます。

前にもお話ししたように、因果関係図とフロー・ダイアグラムとは、使用目的が違います。フロー・ダイアグラムは、問題解決に直結する使い方をします。一方、因果関係図は関係者が、問題が含まれている対象の概念の把握と問題そのものについて理解し、コミュニケーションするための材料として主に使います。したがって、関係者間のコミュニケーションが不要であったり、取り上げる概念が既に明確である場合には、因果関係図は改めて描く必要はありません。このことについては、このテキストの最後に付けている“附録1: 定性的なシステムズ・アプローチ(システムズ・シンキング)”を参照してください。

この段階ではモデルの変数を使って、取り上げた問題をフィードバック・ループやストック、フローといった言葉で表現します。変数の単位をチェックしながら、どの変数がどの変数に関係しているかを考慮しながら、ストックとフローの関係や情報リンクの関係を決定します。実際にこの段階を実行していると、モデルの変数リストにない変数が必要になったり、あるいは決めていた変数名が妥当でない場合も出てきます。その場合には臨機応変に対応します。

なお、この段階までは数式が出てこないのです。例えば練習用の小さなモデルであるなら、紙の上でも行うことができます。

## 5.6 モデルの完成

次に、モデルに数式を与えます。足し算、引き算、掛け算、割り算を基本にして、変数を結び付けます。なお、内生変数の単位は Powersim Studio の中で、自動的に生成し、それが数式の論理チェックになっています。したがって、外生変数とレベルの初期値にだけ単位を設定し、内生変数については、このツールの中で単位を自動的に発生させてください。仮に、与えた外生変数などの複数の単位に矛盾が含まれている場合には、このツールが論理矛盾を指摘してくれます。

数式の設定の事例を考えてみましょう。例えば、在庫管理のモデルでは、在庫というレベルを満たす生産量がフローで与えられます。このフローには2つの変数に関係していて、さらにレベル自身もこのフローに結び付いているはずですが、どのくらい生産すべきかは、在庫の現在量によって決まるからです。2つの変数とは在庫の適正量と生産を調節するのにかかる時間です。それぞれ、適正在庫量と在庫調整時間と呼ばれています。

生産は、個／週とか個／月のように、単位時間当たりの個数という単位で表されます。適正在庫量、在庫、在庫調整時間はこの生産のフローの単位である個／週を生み出すような関係式によって結び付けられなくてはなりません。最も一般的な式は、 $(\text{適正在庫量} - \text{在庫}) / \text{在庫調整時間}$  となります。単位で見れば(個 - 個)／週であり、予想したとおりの個／週になっています。関係式を特定するだけでなく、パラメータの値を選ぶのもこの段階です。パラメータとは、定数、レベルの初期値、グラフ関数(テーブル関数)を指しています。

定数を選ぶときには現実のシステムを考える必要があります。“5.3 時系列挙動図ほかデータの収集”の段階で集めた値から適切な値を推測できるはずですが、大切なことは、設定する全ての値の間で整合性がとれていることです。時間に関する定数は、時間の単位が異なっても構いません。ツールの中ではある単位に自動的に換算されて取り扱われます。

レベルの初期値も現実的なものでなくてはなりません。ただ、最初のデバッグのためには、ある程度は丸めた数字の方が便利ではあります。

グラフ関数、別名テーブル関数もこの段階で定義しておく必要があります。テーブル関数は単純に2つの変数の関係をグラフで表現したものです。変数間の非線形な関係を表現するのに非常に便利ではありますが、モデルの検証においては設定したグラフ関数の精度とその有効性を明確に述べる必要があることを、決して忘れてはいけません。

テーブル関数を作るときは、変数間の関係の特徴を分かっている必要があります。テーブル関数を設定する場合に、まずは既に集めているデータをチェックしますが、それでも不明な場合には、その対象の有識者に次のような質問をしてみると良いでしょう。

[質問]一方の変数の値の増加により、もう一方の変数の値は増加するのか、減少するのか？

- 増加(または減少)はずっと指数関数的に増加(または減少)するのか？
- 最初は指数関数的に増加し後に減少するのか？
- それ以外の関係にあるのか？

## 5.7 シミュレーション

ここまでくるといよいよシミュレーションができる段階です。概念モデルがコンピュータ上にのせられ、すべての変数と方程式が定義されると、モデルのシミュレーションを行ってその挙動を観察することができるようになります。実際にモデルのシミュレーションを行う前に、自分の頭の中でのシミュレーションによって、モデルはどのような動きをするかを想像してみると良いでしょう。

その予測した結果を頭に描いた状態で、シミュレーションを実行します。よくあることですが、結果が予測したものと違っている場合は、モデルの妥当性を追及するスタートポイントが与えられたこととなります。それは、モデルの構造が間違っているのかもしれない、ある変数を間違えたのかもしれないし、自分自身の予測が間違っていたのかもしれないなどの追求を始めます。

コンピュータ・シミュレーションでは、以下の3つのことに注意する必要があります。

### (1) モデルのシミュレーションを行う期間

これはモデルの目的によって異なります。

Powersim Studio ではデフォルトが1年間になっています。

カレンダーの期日を使って計算開始時刻(Starttime)と終了時刻(Stoptime)とを設定するのがデフォルトになっていますが、カレンダーを使わないで“計算回数”などにより、計算期間や次に述べる刻み時間を設定することも可能です。

### (2) シミュレーションの刻み時間

システム・ダイナミクスは、映画のフィルムのコマのように、一定の時間ごとに挙動を計算し

て映像を記録しています。そのコマの時間間隔を刻み時間(タイム・ステップ:timestep)と呼んでいます。

この間隔が短ければそれだけ多くの計算を行い、シミュレーションに時間がかかります。デフォルトは1日です。もし、シミュレーションを実行していてその挙動に振動が起るようで、かつ、それが不自然であると判断したら、このタイム・ステップを短くして、再シミュレーションを行う必要があります。

### (3)シミュレーションにおいて採用している積分法

シミュレーションでは積分を使って方程式を解いています。数値積分法としては、オイラー法(Euler Method)と次数の異なるルンゲ・クッタ法(Runge-Kutta Method)が、Powersim Studioには準備されています。曲線を階段状に分割して積算するオイラー法がデフォルトです。ほとんどのシミュレーションは、オイラー法で問題なく解くことができますが、仮に、モデルが振動を起すようであれば、オイラー法のままでタイム・ステップを短くするか、あるいはタイム・ステップはそのままにして、ルンゲ・クッタ法に切り替えてみると良いでしょう。ルンゲ・クッタ法は積分すべき曲線をそれに近い曲遷に近似して積分していますから、オイラー法のようなぎざぎざの計算誤差が出ませんので精度良く計算できる可能性があります。

上記の三つの条件を設定して、いろいろな条件のもとでシミュレーションを行い、結果を確認します。モデルをよく理解するためには、モデルの構造とシミュレーションの結果とを結び付けて考えることが大切です。

期待するシミュレーション結果が得られないときは、モデルの構造を再分析し、なぜそのような結果になるのかを調べる必要があります。この場合に、フロー・ダイアグラムを因果関係図と見立てて、要素間の因果関係について再び分析します。その際、1タイム・ステップずつの計算を行い各要素の値を確認しながら、因果関係の分析を進めます。

期待していない結果の原因が分かったら、モデルの構造のどこを変えれば正しい動きをするのかをモデルを使って実験的に確認します。

モデルが現実を適切に表現するものになったら、モデルの妥当性を検証します。モデルの妥当性が確認できた後に、最終目標である、例えば、現状のシステムの最適化やリスク評価などを実行するための戦略的シミュレーションのステップに進みます。

以上

## 附録1 : 定性的なシステムズ・アプローチ(システムズ・シンキング)

システムズ・シンキングは、定性モデルを使ったシステムズ・アプローチです。これだけで具体的な経営問題を解決するには力不足の感じを否認ませんが、システムズ・アプローチに馴染みの薄い人たちにも分かりやすい表記法であることが大きな特徴になっています。

特に、対象としているシステムに対して関係者の価値観が様々であり、問題解決に当たるにしてもその問題自身に対する認識が一致していないような場合には大変有効な方法論になります。このような場合には、いわゆるソフトシステム・アプローチを実施する必要があります。そのために各人のメンタル・モデルを視覚化する手段として、“因果関係図”と“時系列挙動図”とがあります。因果関係図は、定量モデルの表記法である厳密なフロー・ダイアグラムと違って、コミュニケーションの種類に合わせてその粗密度を任意に設定できることが、多くの人たちへの分かりやすさとなっています。したがって、画一的な因果関係図を描くのではなく、使おうとしているコミュニケーションの状況にフィットした粗密度の因果関係図を描く必要があります。

このテーマについては、POSY 社のホームページに設けている“SD 閑話”に上記のタイトルで記事を掲載していますが、その一部を抜粋して以下に収録します。

SD 閑話の記事は以下の URL でご覧下さい。

<http://www.posy.co.jp/kanwa3-A602-f.htm>

SD閑話-3 定性的なシステムズ・アプローチ

### 1. 定性モデルの表記ツール

我々はモデルと言う単語をいろんな状況で使います。モデルは、実物ではなく、実物の特徴を表現した偽者です。さて、システムズ・アプローチでは、その偽者をどのように使いたいかというと、実物を代表するある特性が、その特性自身と外部の条件変化とによって、時間的にどのように変化するかを見たいのです。しかも、その実物の一部分だけがどのように変化するかではなく、全体がどのように変化するかを見たいのです。

システムズ・アプローチ by JWF(フォレスター教授によるシステムズ・アプローチ)では、そのような時間的変化を定性的に把握するために定性モデルを使います。そのモデルの表記法として、因果関係図と時系列挙動図とが準備されていることは既にお話しました。時系列挙動図とは、そのモデルに含まれる主要な要素が、時間経過と共に、どのように変化するかを予想する図面ですからツールとしては簡単で、後で少しだけ触れることにします。以降では因果関係図について詳しくお話しします。

システムズ・アプローチでは、“モデル”の概念を次のように考えています。

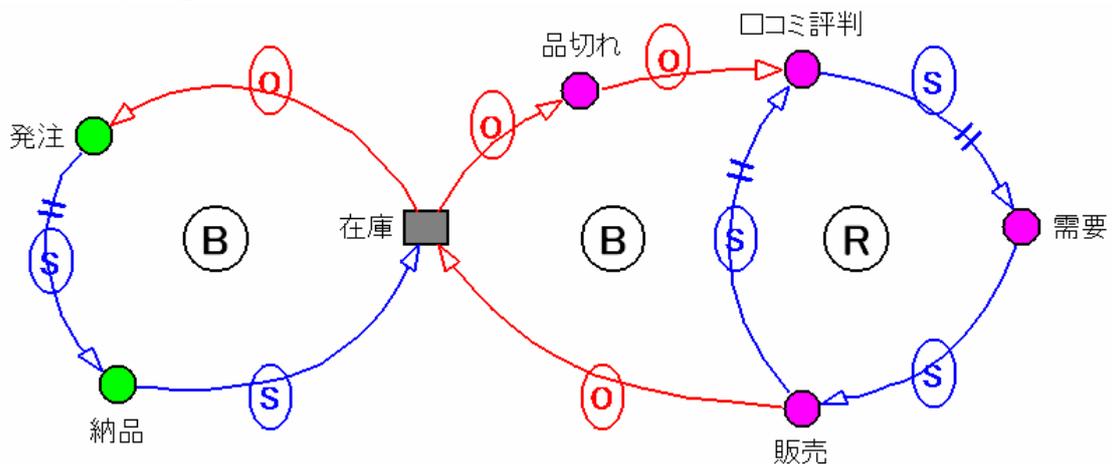
- ① 対象としているシステムを、注目している視点から眺める。
- ② 捨象と抽象により、その視点における本質的な“要素とその関係”を抽出する。
- ③ その“要素とその関係”を再合成して得られるシステムがモデルである。

システムズ・アプローチにおける関係とは因果関係のことです。原因としての要素が大きくなると、結果としての要素がどうなるかという関係です。原因に対して結果が同じ傾向に変化する場合を、“Same Direction”と呼び、逆の傾向に変化する場合を“Opposite Direction”と呼びます。原因から結果に向けて両者を矢印で結びますが、Same Direction は青色に、Opposite Direction は赤色に着色するのが一般的です。矢印の傍に、Same Direction は S や + を、Opposite Direction は O や - を描くこともあります。因果関係が比較的短時間のうちに現れる場合と、原因に対して結果が時間をおいて現れる場合とがあります。後者の時間遅れがある場合には、矢印の上に短い線を二本加筆します。

さて、商店経営に関して、仕入れと販売に注目して、主だった要素を選択し、それら二者間の因果関係を以下に描いてみます。

- 販売 → 口コミ評判
- 口コミ評判 → 需要
- 需要 → 販売
- 販売 ⇒ 在庫
- 在庫 ⇒ 品切れ
- 品切れ ⇒ 口コミ評判
- 在庫 ⇒ 発注
- 発注 ⇒ 納品
- 納品 ⇒ 在庫

二要素間の因果関係をさらに繋いでいくと、下方の図になります。これを因果関係図 (CLD: Causal Loop Diagram) と呼んでいます。



上図の中には、矢の先がその前の矢の元に連なって、それを辿るうちに元の位置に到達するループが3つ存在します。フィードバック・ループと呼びます。その中に、丸Rとか丸Bとか描いてあります。丸Rは Reinforcing Loop で、一定の方向に強化(負の強化もあります)されることを、丸Bは Balancing Loop で、いずれかの値に収斂することを意味しています。

原因と結果の傾向が逆になる繋がりである Opposite Direction がループの中に奇数個あると、ル

ープを巡るごとに、要素の大小が交互に変わりますから、それらの要素の値は一定の値の周りに収まります、したがって、Opposite Direction が奇数個あれば Balancing Loop になります。

商店の経営者は、自分の仕事を仕入れと販売の視点で見つめるなら、この因果関係図に近い仕組みで商店経営を行なっていると認識するはずですが、すなわち、経営者は頭の中にメンタル・モデルとしてこのような構造を蓄えているのです。したがって、因果関係図は関係者のメンタル・モデルを目に見える形で表現するための手段でもあるのです。

ある問題に複数の人達関係して議論を進めるうちに、互いに相手の考えが分かったつもりになったが、次のステップに進んだところ、実は理解が食い違っていたというようなことがよく起こります。それは、互いのメンタル・モデルを理解していたのではなく、表面的な言葉だけを聞く人が都合の良いように解釈していたに過ぎなかったからではないでしょうか。情報交換が曖昧すぎたのです。因果関係図のように互いに目に見える形で表現して、それを眺めながらその周りで議論するなら、明確な共通認識のもとで互いに許容できる範囲に結論を導くことができるようになります。

このことは、複数の関係者間の問題解決だけではありません。我々が文章をまとめるときに、構想を練って、一先ず文章化して、推敲を重ねます。さらに、時間をおいて読み直します。この過程で、最初に想定していた結論とは異なる結論に気付いたり、論理の自己矛盾に気付いたりします。実は、定性モデルを描くときも、全く同じ現象が現れます。我々は、定性モデルを描くときには、往々にして現状の AsIs モデルから描くことが多いと思います。その AsIs モデルを描いた後でそれを眺めて、現状の問題を解決するために、付け加える要素はないか、害となっていて削除すべき要素はないか、どこかにリンク線を加えたり、あるいは既存のリンク線を削除すると問題解決に向かうのではないかなどを検討します。

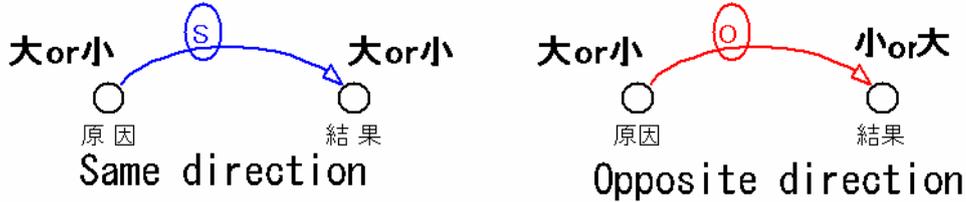
しかし、実際にやってみると良く分かるのですが、現状の AsIs モデルを描いている途中で、その業務プロセスなどのモデルの構造の不具合さに気付きます。自分の頭の中のメンタル・モデルを視覚化することで、自分自身が自分のメンタル・モデルで誤魔化してきた事実気付くようなのです。したがって、定性モデルはコミュニケーション・ツールであると同時に、自己学習のためのツールでもあるのです。

ところで、現状の AsIs モデルを描いている途中で、改善策を次々に思い浮かべると話しましたが、そこでは辛抱してその場で AsIs モデルを中途半端に変更しないことです。思いついた内容はメモをとるだけにして、AsIs モデルは AsIs モデルとして完成させ、その全体を眺めた上で、改造あるいは革新すべき点を総合的に議論すべきだと言うことを付記しておきます。

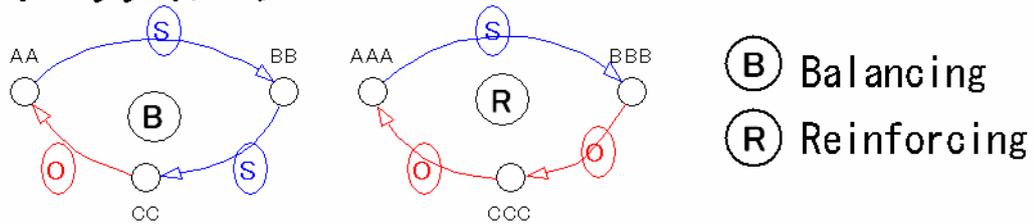
さてここで、因果関係図の表記法をまとめておきましょう。要は、対象としているシステムの因果関係が表現できればいいわけですから、描き方は色々あります。ここでは、パワーシム社(Powersim Software AS:ノルウェー)が提供している Studio 8 を使って、因果関係の表記法と、フィードバックループの表記法とを以下に記します。このツールは、本来、定量モデルを構築するために準備されているのですが、編集がやり易いことから、定性モデルの表記においても都合よく利用することができます。なお、きれいで分かり易い因果関係図を描くには、上記のツールの類が便利だとは思いますが、あくまでも因果関係が描ければ良い訳ですから、手描きでも、あるいは汎用のお絵

描きソフトでもいっこうに構わないことは言うまでもありません。

### 因果関係図 (CLD)



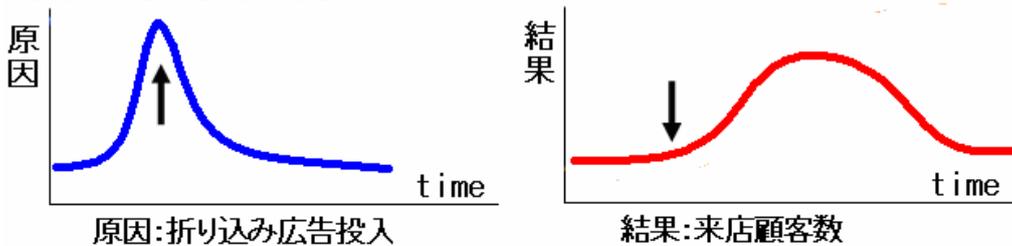
### フィードバック・ループ



次に、時系列挙動図について少しだけ説明しておきます。例として、商店の経営者が時々仕掛ける新聞折り込み広告などの販売促進活動について考えて見ましょう。商店主は、新聞に折り込み広告を入れたり、店頭キャンペーンの飾り付けをするために費用をかけます。その結果、需要が喚起されて商品販売が上向きますが、それにはある程度の遅れ時間(リードタイム)がかかることを商店主は知っています。では、商店主は原因の販売促進費用と結果の販売額とが時間的にどのように変化すると考えているでしょう。もちろん、その量的な判断もしているわけですが、その前に販促費用をかけると、どのタイミングで販売が上向き始めるかという時間的な相互の関係が重要になってきます。そのタイミングを見誤りますと、せっかく販促したにもかかわらず品切れを起こしたり、在庫の山を築いたり、あるいは資金不足に陥ってしまいます。

時系列挙動図は、対象としているシステムの主要な構成要素が時間的にどのように変化するか、あるいは将来にわたって変化して欲しいかについて、原因と結果のタイミングについて注視して描く挙動図です。いずれの図でも時間軸は共通で、縦軸の変数値は傾向を示すだけで数値としては意味をなしません。もちろん、原因については、関係者のメンタル・モデルからではなく、既成の事実に基づくものもありますので、その場合にはその定性的な時間経過の傾向を原因要素の時系列挙動図として採用します。

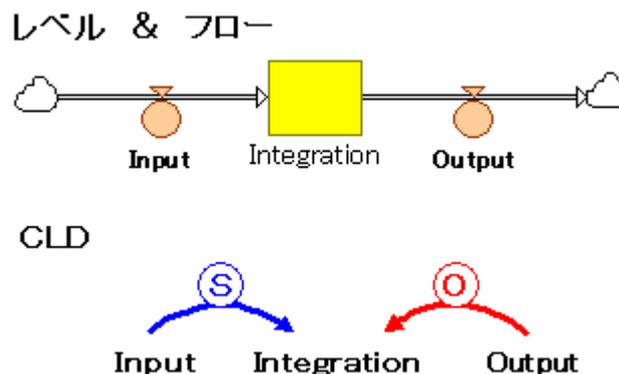
### 時系列挙動図の例



### 2. 因果関係図は必ず描く必要があるのだろうか？

フォレスト教授が 1960 年ごろ発表された論文“Production Distribution System”に触発されて、

1961年に当時の通産省からMITに留学され、帰国された後にシステム・ダイナミクスを使って学位論文をまとめられた日本の先達がいらっしゃいます。フォレスター教授がシステム・ダイナミクスを創案されたのが1950年代半ばと言われていますから、日本人がシステム・ダイナミクスに取り組んだのは随分早かったわけです。そのことは1972年に“成長の限界”として発刊された“人類の危機に関するプロジェクト”を推進したローマ・クラブの活動において、日本国内の産業界と学会とが協力してローマ・クラブ日本チームを形成していたことから伺い知ることができます。社会の動きはシステム・ダイナミクスに追い風となっていたようですが、日本における計算機技術のレベルの低さは、足かせとなり、システム・ダイナミクスに基づくシミュレーションの実行は大変だったそうです。当時、卒論や修論を、システム・ダイナミクスを使ってまとめられた方々のお話を伺うと、指導教授にお願いして企業の汎用コンピュータを夜間に使えるように準備してもらい、一晩に一ケースのシミュレーションの実行がやっとできたとのことでした。プログラム言語は、ダイナモ、ベーシックあるいはフォートランです。したがって、数式をプログラム言語で記述するだけですから、モデルの構造を視覚化するための因果関係図は不可欠だった訳です。しかし、現在のシステム・ダイナミクス・ツールは、図式化されていますから、ツールを使って描いた構造図であるフロー・ダイアグラムが詳細レアーの因果関係図とも言えるのです。具体的には、情報伝達のリンク線はもちろん、レベルとフローの関係も、下図に示すように因果関係を表しています。



上記を理解した上でシステム・ダイナミクスの定量モデルであるフロー・ダイアグラムを眺めると、そのまま因果関係図と見なすことができます。したがって、関係者間のコミュニケーションに因果関係図などを使う必要がないのであるなら、フロー・ダイアグラムのほかに因果関係図を描くことはないと思います。

### 3. 定性モデルの具体的な構築法

1999年にロンドンのとあるホテルの会場で、Powersim社の講習会を受講しました。講習2日目の朝、講師が当日の新聞を配って、皆で討論しながらこの記事进行分析して因果関係を描いてみましょうと話しました。当日の新聞にはタラ魚の不振と漁船の規制に関する記事が掲載されていたように覚えています。私はこのワークに大変新鮮な感じを受けましたので、帰国してから毎朝新聞を見ては、どれかの記事を定性モデルとして描いてみました。結果としてこれはなかなか良い方法

で、新聞情報を漫然と読むのではなく、その内容を構造的に捉える訓練になったように思います。最近では新聞を読みながら、紙に定性モデルを描いてみることはありませんが、頭の中で問題を構造的に捉えることができているように思います。皆さんもやってみませんか。

さて、定性モデルを描こうとするときには、以下に記載するようなステップを辿ると都合が良いと思います。また、取り上げる対象としてご自分が取り組んでいる問題が、ビジネス・プロセス・モデルの場合には、そのAsIsモデルの構築から始めると取り組み易いはずですが、ただ、訓練として取り組むには興味を感じた新聞記事を対象として始めても構いません。

- ① 問題は何か？
- ② 問題をどのように認識しているか？
- ③ 分析を始める貴方の立場は？
- ④ 貴方の分析の目的・狙いは？
- ⑤ 予想している解決策は？
- ⑥ 問題に関係する要素は？
- ⑦ それぞれの2つの要素間の関係は？
- ⑧ 定性モデル(因果関係図・時系列挙動図)を描く。
- ⑨ 各自のメンタル・モデルを最大限に引き出して、現在の状況を分析し、問題点を明確に表現する。
- ⑩ その問題を解決するための、リンクの加減、ループの加減、要素の加減等を検討したうえで、定性モデルを参照しながら、問題解決のシナリオを創出する。

最後に、Studio 8 を使って因果関係図を描く方法(前述の項目⑧)を詳細に記します。

- ① 目的の視点から問題(対象)を眺め、要素を抽出し、SD モデルの適当なシンボルを最小化して、それに抽出した要素の名前を名詞で付けて並べる。  
錯綜を避けるために負のイメージの名前は付けない。Ex.利益減少、需要減など。
- ② 似かよった要素をグルーピングして色分けする。
- ③ グループごとに集めて、因果関係を考えてリンク線を引く。  
正方向は青色、逆方向は赤色を付ける。
- ④ 多分、スパゲッティ図になる。釣り糸のもつれを解くように要素を動かす。  
要素に付いているリンク線は外れることはない。
- ⑤ 交差が残り、許せない場合には、CLD の分割を先ず検討する。  
簡易的に交差を除くには、リンク線が交差する原因となっている要素のダミーを作って配置する。そのダミーの名前は、元の要素の名前の後ろに数字をつけるなどして、ダミーであることが分かり易いような工夫を施す。
- ⑥ CLD ができたら、デザインモードのアイコンをクリックして OFF にし、警告マークの表示を消す。
- ⑦ 他のアプリケーションで、できあがった因果関係図を使う場合には、イメージ・コピー機能を使ってコピーして、他のアプリケーション上にペーストする。

以上

## 附録2 :Ps Studio 8 Demo/Express

### 1. 評価版 Ps Studio 8 Express と Demo の特徴

2種類のフリーの評価版を提供しています。

(1)Express 版は、SD に全くの初心者に向いています。SD の概要を学習して SD の実用化の可能性を評価するためにお使い下さい。

ただし、Express は Professional 版と同機能なので、最適化とかりスク分析の機能は使用できません。

(2)Demo 版は、商用版の最上位の Enterprise 版と同じですから、最適化とかりスク分析の機能は勿論のこと、SAP BI データベース、Oracle データベースなどとも接続できます。

項目	評価版 Ps Studio	
	Demo	Express
相当商品版	Enterprise版	Professional版
構築要素数	制限なし	50個
有効期間	30日	60日
同一PCへの再インストール	不可	可
有効期間終了後のモデルのロック	ロック(*)	ロックなし

(\*)商品版をインストールして解除

### 2. 評価版 Ps Studio 8 Express と Demo のダウンロード

ダウンロードするためには、以下の URL から評価版のダウンロードのページに入ってください。

- ① <http://www.posy.co.jp/PS-download-f.htm> を開く。
- ② 上端の“評価版のダウンロード”をクリック。

#### (1)Express 版のダウンロード

Studio Express のダウンロードの空色のボックスをクリックすると、Powersim 社の Express のダウンロードのページに入りますから指示に沿って進んでください。

#### (2)Demo 版のダウンロード

Studio Demo のダウンロードのオリーブ色のボックスをクリックすると、“Download Studio products”のページに入ります。

その“Download demo of Studio 8 Enterprise”をクリックして、指示に沿って進んでください。

以上