DBS-08-02

因果ループからSDモデルを構築する方法について

– システム思考８基本型とことわざの考察 –

山口　薫, Ph.D.
同志社大学大学院ビジネス研究科
福島　史郎, MBA
同志社大学大学院総合政策科学研究科博士課程

1　はじめに

システムダイナミックス（System Dynamics, ＳＤ）はＭＩＴのジェイ・フォレスター (Jay Forrester) 名誉教授によって約５０年前に提唱された社会科学分野におけるコンピュータモデリングを利用した問題解決の手法で、フィードバックによる情報の流れやシステムの遅れ、システムの境界を明示化するダイナミックモデリング手法である。モデリングの分野も、ビジネス経営戦略、経済学、公共政策、環境問題等多岐にわたり、まさに現代社会が必要としている学際的ニーズにも合致する。２００６年にはＳＤ創始者のフォレスター名誉教授が、オペレーションズリサーチ科学で最高の名誉である IFORS' Operational Research Hall of Fame に殿堂入りした。これを機会にＳＤの認知度が急速に広まり、ＳＤによるシミュレーション手法の研究が今後ますます社会科学の分野でも盛んになってくるものと予想される。

システムダイナミックスは、複雑なシステムの相互依存関係を、ストック、フロー、変数、矢印と呼ばれる４つのアイコンを組み合わせて、お絵かきに似たような感覚でモデル構築してゆくことを特徴としている。このように一見簡単に見える手法ではあるが、完成した SD モデルを一般の読者に平易に説明するのは至難の業のようにみられた。

そこで複雑に見える SD モデルの変数の相互依存・フィードバック関係をさらに直感的にわかりやすく説明できるコミュニケーション手法として

・著者の許可なく、本ディスカッション・ペーパーからの転載、引用を禁じます。
・ＤＢＳディスカッションペーパー・シリーズは、オムロン基金により運営されています。

１９７０年代頃から、因果ループ図 (Causal-Loop Diagram, CLD) が提案され始めた。勿論こうした定性的な因果ループ図でシステムの構造をとらえることによる論理の落とし穴も Richardson [3] 等によって指摘された。にもかかわらず、小学生にも理解できるような直感的なこうした方法は、SD の定性モデル (Qualitative Model) として広く利用されるようになった。

こうした SD の定性モデルをポピュラーにしたのが Senge[5, 6] である。同書は、因果ループによるシステム思考を第５番目の規範 (the Fifth Discipline) としながら、ビジネスにおける組織と人間の行動、学習する組織を論じたものであるが、意外にも学校の先生方にも広く読まれた。同書を契機に、この因果ループ図による SD の定性モデリング手法は、システム思考 (Systems Thinking) として広く利用されるようになった。例えば、Kim and Anderson [2] は、経営コンサルティングセミナー等で広く用いられている。８０年代のマッキントッシュの登場とともに、難解な SD 言語をユーザーフレンドリーなお絵かき感覚で使用できるようなソフト STELLA が開発されてこうした傾向に拍車がかかった。このソフト開発者である Richmond [4] も広く読まれている。国内では、西村 [10] が、システム・シンキングの思考法をビジネスに於ける様々な事例を紹介しながら、平易に解説している。このように、因果ループ図を応用しながらシステムの構造を理解し、問題解決に応用してゆく傾向が一般的となりつつある。

しかしながら、SD の代表的なテキスト [7] で、MIT のジョン・スターマン教授が指摘しているように、経営戦略の立案や組織改革においてソリューションを探索する場合、因果ループだけで十分な知見が得られる場合は少ない。因果ループを更に発展させてシステムダイナミックスモデル（以下、ＳＤモデル）を開発して、パラメータを変化させながらシステムの振る舞いを観察することがより深いシステム構造の理解につながり、ひいては有効的、実践的なソリューションに到達できると考えられる[1]。

それでは、因果ループ図（一般的には SD 定性モデル）から SD（定量）モデルを構築するにはどうすればいいのだろうか。因果ループ図からＳＤモデルを構築するための一般的法則はなく、こうしたモデル構築作業は専門的知識を有するＳＤモデラーの腕の見せ所のように思われてきた。そこには論理的に説明できない思考のジャンプがあるようにもみなされてきた。本論文は、まずこうしたジャンプを容易にするモデル構築プロセスの一般的手法を提案し、次にシステム思考８基本型（以下に説明）にこの手法を応用し、その有効性を論じることを目的とする。またその課程で、システム思考８基本型の持つ意味を、東洋思想の知恵の結晶であることわざと関連させながら探索してゆくことにする。

[1] ロンドンビジネススクールの Kim Warren [8] のように、因果ループ図を援用しないで、数個のストック・フローからモデルを構築してゆく方法の方が効率的であるとする見解もある。筆者の一人もこうしたモデル構築の有効性、効率性に賛同し、SD の基礎理論を学んだ MBA 学生対象のビジネスモデリングという科目では、同書の方法論に沿って講義を展開している。

2　SD モデルの構築プロセス

2.1　因果ループ図の作成

因果ループ図を作成するには、まず解決したい問題点を明確にし、そうした問題点を排出していると考えられるシステムに関するキーワードを集めることから始める。そしてこれらのキーワードを、原因－結果の対として整理する。原因 (Cause) の増減が同じ方向に結果 (Effect) の増減をもたらす場合は、正の因果関係があると判断し、青色の矢印で結ぶ。

原因の増減が反対の方向に結果の増減をもたらす場合には、負の因果関係があるとして、赤色の矢印で結ぶ。図１は、１５のキーワードから９つの因果関係を見いだした図を表している[2]。

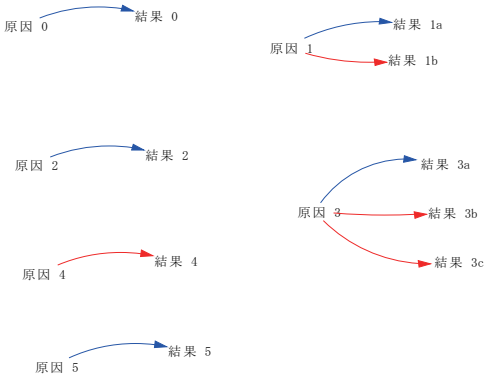

図 1: キーワードの因果関係

次の作業は、こうして見いだした因果関係の中から、ある因果関係の結果が他の因果関係の原因となっているものを見つけ出し、同じ法則で青、または赤の矢印で結ぶことである。図２（左）は、このようにして新たに３つの因果関係が結ばれ、一つのループが完結した図を表している。

こうして出来たループの赤の矢印（負の因果関係）の数が偶数個（ゼロ個を含む）の場合に、このループを正のフィードバックループ (Positive Feedback Loop) または増強ループ (Reinforcing Loop) と呼び、奇数個の場合には、負のフィードバックループ (Negative Feedback Loop) または平衡ループ (Balancing Loop) と呼ぶ。図２（左）で見出されたループは負の因果関係がゼロであるので、正のフィードバックループであることを表している。正のフィードバックループは、システムを指数的 (Exponentially) に増強させるパワーを持っており、成長のエンジンの役割を果たす。

図２（右）は、さらに負のフィードバックループが一つ見いだされたことを表している。負のフィードバックは、システムを安定させる作用をする。即ち、システムが均衡（平衡）状態から乖離したとすれば、元の均衡状態に引き戻すような安定化動作をするのである。

こうしたループ探索作業をすべての因果関係がループとして完結する (Closign

[2]本論文における因果ループ図、および SD モデルはすべてフリーソフト VensimPLE を用いて作成した。同ソフトは、http://www.vensim.com からダウンロード出来る。SD モデル作成用ソフトは、これ以外にも STELLA, iThink, PowerSim 等があるが、因果ループ図作成には VensimPLE が最適である。事実、他のソフト利用者も因果ループ作成には VensimPLE を多用しているようである。

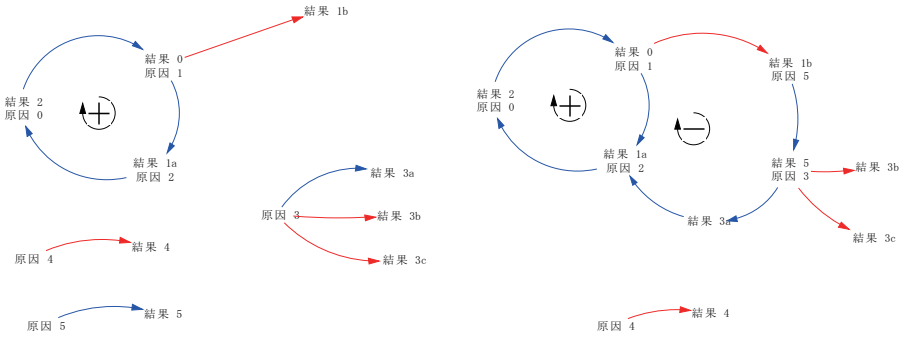

図 2: 因果ループ（左）のその拡張（右）

the Loop) まで続けてゆく。といっても対象としている問題点の分析にとって明らかに不適切と思える因果関係をすべてループにこじつける必要はない。このことは、対象としている問題を考察するためのシステムの境界 (Boundary) を、どこに引くかの問題に帰着する。

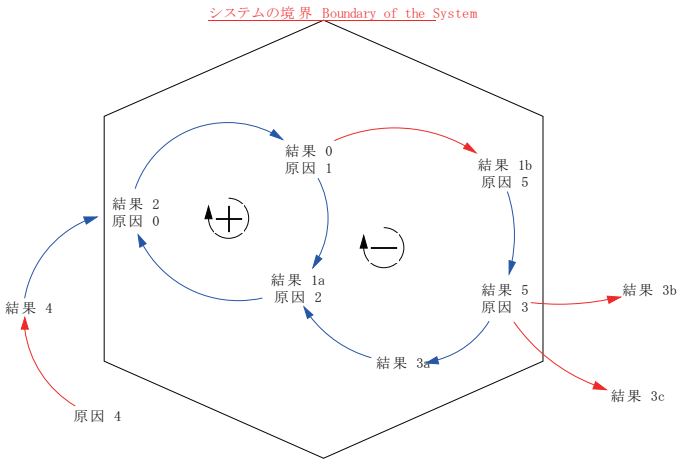

図 3: 因果ループ図

因果関係の原因や結果がシステムの外から来ていると考える場合が適切な場合もあるし、またシステム内の因果関係の結果が、システムの外に出て行き、フィーバックしない場合もある。図 3 は、こうして完成したシステムの境界を明示した因果ループ図である。同図には、正と負のフィードバックループがそれぞれ一つずつある。システムの動きが増強に向かうのか、または均衡（停滞）に落ち着くのかは、こうした 2 つのループのせめぎ合いで決まる。これが因果ループを用いた定性的な SD モデルによる構造分析である。

2.2　人口の因果ループ図

それでは上で考察した因果ループ図の作成プロセスを、人口問題に当てはめて具体的に考えてゆこう。図 4 は、最も簡単な人口増減の因果関係を表す。出生が増加すると人口が増加するという正の因果関係と、死亡が増加すると人口が減るという負の因果関係が示されている。

次に因果関係から因果ループを探してゆく。具体的には人口という結果がさらに他の原因となっている因果関係を見出すことである。まず、人口の増加が更なる出生の増加につながる正の因果関係、そして人口の増加がさらなる死亡の増加につながる正の因果関係を新たに見出すことができる。その結果、出生－人口－出生という因果ループと、死亡－人口－死亡という因果ループが見出される。図 5（左）にこれらの因果ループを示す。

さらに、その因果ループが正のフィードバックループであるのか、負のフィードバックループであるかを特定してゆく。すでに上述したように、負の因果関係（赤い矢印）が偶数個ある因果ループは、正のフィードバックループとなり指数的人口増加を生み出す構造となる。他方、負の因果関係が奇数回あるものは、負のフィードバックループとなり、人口の均衡、安定化をもたらす構造となる。図 5（右）で図示しているように、出生－人口－出生という因果ループは正のフィードバックループであり、死亡－人口－死亡という因果ループは負のフィードバックループとなる。

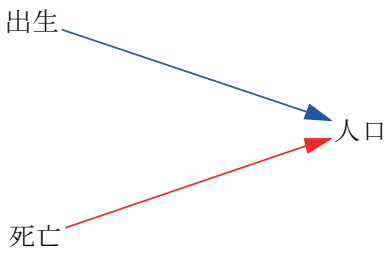

図 4: 因果関係

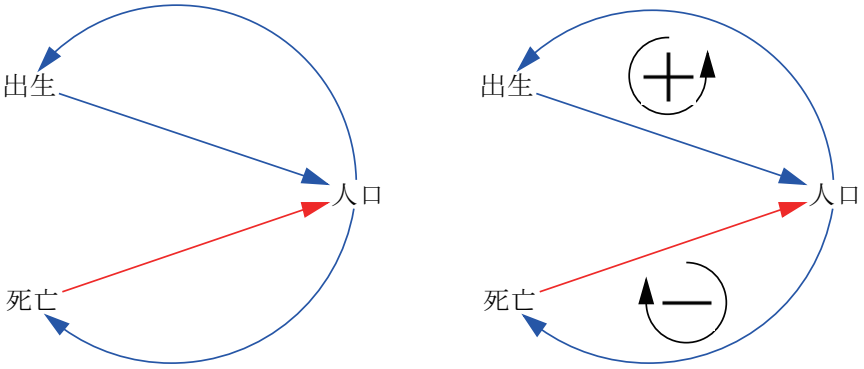

図 5: 因果ループ（左）と正・負のフィードバックループ（右）

このようにして、人口動態システムは、各々一つずつの正と負のフィードバックループから構成されていることが判明した。この正と負のフィードバックループの力関係が、人口増加、または縮小をもたらすのである。以上が、

システム思考によるシステム構造の簡単な分析例である。

それでは、出生や死亡がどのように変化すれば、人口が増減するのであろうか。システム思考による定性モデルでは、こうした定量問題に答えることが出来ない。SD 定性モデルの限界である。こうした問題に定量的に答えるためには、SD モデルの構築が不可欠となる。

2.3 SD モデル構築プロセス

因果ループ図を眺めていると、ある論理矛盾が内在していることに読者は気づくだろう。上の例をとると、人口が出生数を決め、それが人口を決定する、すなわち人口が人口を決めるという恒等式的な同時方程式体系になっているという点である。これを回避するためには、例えば昨年度の人口が今年度の出生数を決め、それが今年度の人口を決めるといった時間的経過の導入が不可欠となる。こうした緩衝帯の役割を果たすのが、SD モデルではストックと呼ばれるものである。従って、因果ループが有意となるためには、そのループの中に最低限一個のストックが存在しなければならない。この例では人口がストックとなる。図 6 に於いて、ストックは外部を四角形で囲んだ形で表現している。

因果ループ図では、原因と結果の関係にある事象を表記するだけであり、ストックという観点では区別して表記していない。SD モデルを成立させるためには、少なくとも一つのストックが必要となる。従って、因果ループから SD モデルを構築するにあたって、まず最初に行うべき作業は、因果ループに表示された事象からストックを特定するということである。

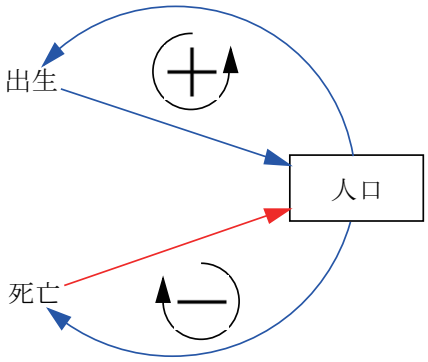

図 6: ストック明示の因果ループ図

システムダイナミクッスでは、システムはストックの集合として表現される。そのストックを増減させるのがフローであり、フローのみがストックを増減させる。即ち、ストックとフローは不可分な関係であるとするのが、SD によるダイナミックスの考え方である[3]。よって因果ループからストックを特定できれば、次にストックを増減させるフローを見つけなければならない。

システムの中で、ストックとフローを見分けるためには、時間を停止する思考実験が有効である[4]。なぜなら、フローは時間が停止したとき消失するが、

[3]時間とストックの関係、ストックとフローの不可分な関係等システムダイナミックスの基礎理論的概念に関する詳細な分析については Yamaguchi[9] を参照のこと。

[4]但しこの思考実験は万能ではない。マクロ経済学における国内総生産（GDP）は 1 年間にお

ストックは依然として存在し続けるからである。今、毎秒１リットルの水が水道の蛇口からお風呂に注ぎ込まれているというシステムがあるとする。仮に時間を停止すると、水道の蛇口から流入する水量はゼロになる。従って水道の蛇口から流入する水量はフローであると判断できる。しかしながら、仮に時間を停止してもお風呂の水は依然として存在し続ける。従って、お風呂に溜まった水はストックであると判断できる。

図５の因果ループでこのことを再確認すると、人口がストックであると特定できる。なぜならば、時間が仮にある時点で停止したとしても、その時点における人口は確実に存在すると考えられるが、時間が停止すれば出生も死亡も発生しなくなると考えるのが自然であるからである。

ストックが特定できたら、そのストックを中心としてそのダイナミックスを分析していく。このためには、ストックに入ってくる矢印が重要である。なぜなら、入ってくる矢印がそのストックの振る舞いに対して影響を与えるからである。図６では、ストック「人口」に入ってくる矢印としては、出生からのものと、死亡からのものがある。まず、出生からの矢印に着目する。出生からの矢印は正の因果関係である。このように正の因果関係を表す出生はさらなる人口の増加につながるから、出生が人口に対してインフローとなるように記述する[5]。まず人口というストックに対して出生という名前のインフローを合体させ、次にストック「人口」からインフロー「出生」を矢印で結ぶ、という手順で簡単にこれらの作業を進めることができる。ここまでの操作を行ったものを図７（左）に示す。

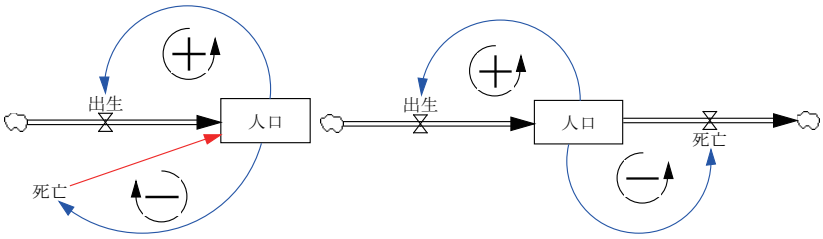

図 7: インフロー（左）とアウトフロー（右）に置換

次に、死亡からの矢印について考える。死亡からの矢印は負の因果関係である。即ち、死亡の増大は人口を減少させることになるから、ストックからのアウトフローとして記述する。人口から死亡というアウトフローを流出させ、次に人口からアウトフロー死亡を矢印で結ぶ、という手順で作業を進める。この操作を行ったものが図７（右）である。

ける総生産量という意味で明らかにフローであるが、GDP をストックとし GDP の変動量をフローとするモデリングも可能である。さらに、距離と速度はストック・フローの関係にあるが、フローの速度をストックとし、その加速度をフローとするモデリングも可能である。これは２階の微分方程式を１階の連立微分方程式として扱う手法と同じである。

[5]モデルの具体的記述に関しては、フリーソフト VensimPLE に添付のマニュアル Vensim 5 User's Guide や日本未来研究センター (www.muratopia.org/JFRC) に掲載の入門マニュアル等を参照されたい。

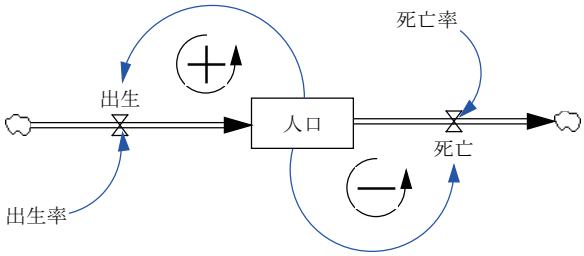

図 8: 数値データＳＤモデルへの拡張

このようにして出来上がった SD モデルのフレームワーク（枠組み）に対して最後に魂を吹き込むのが、各変数への数値データの入力である。この作業は SD モデリングに対する多少の熟練を必要とする。ここでの簡単な人口モデルに関しては以下のようになる。ストック人口とそのフローである出生、死亡は不可分となるので合体図を作成した段階で (VensimPLE では) 式が自動的に作成される。よって人口が出生や死亡を決める式を定義すればモデルは完了する。そのために、出生率、死亡率という定数を新たに導入し、ＳＤモデル図 7（右）を、図 8 のように拡充する。

VensimPLE によるプログラミングコードは表 1 となる。ここで、INTEG

表 1: 人口の SD モデル式

人口 = INTEG (出生 - 死亡, 100)
出生 = 出生率 * 人口
出生率 = 0.03
死亡 = 死亡率 * 人口
死亡率 = 0.01

は VensimPLE の表記法による積分関数を表し、その第１引数（ここでは出生－死亡）は被積分変数、第２引数（ここでは 100 ）は、ストック人口の初期値を表している。図 9 に SD モデルのシミュレーション結果を示す。出生率が３％、死亡率が１％で人口が変化するとすれば、即ち純出生率が２％で人口が純増するとすれば、初期値１００の人口は、１００年後に７２４．４６になることを示している。

2.4　ストック同士が隣接する場合

因果ループでストックを２つ以上見いだした場合、ストック同士が隣接する場合がある。図 10（左）にストック「人口」とストック「環境」が隣接する例を示す。

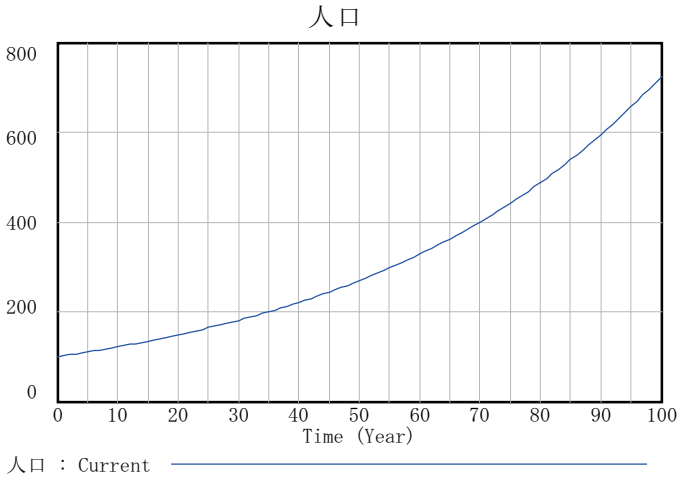

図 9: 「人口モデル」のシミュレーション結果

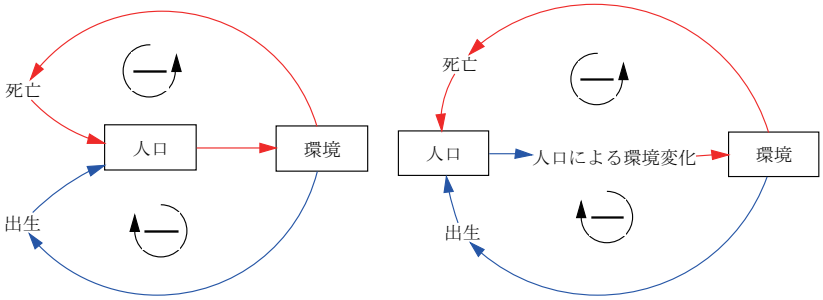

図 10: 隣接するストック同士（左）の間に要素を追加（右）

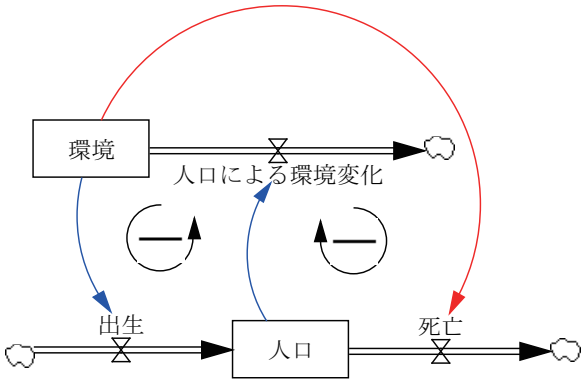

図 11: ＳＤモデルの構築

こうした場合には、ストック「環境」に入る矢印を考えるにあたって、「人口による環境変化」といったストックとならない事象（より具体的にはストッ

ク「環境」のフローとなる変数）を追加する。追加後の因果ループ図を図 10（右）に示す。このように修正した後に、ストック「環境」について上述の作業を適用すると、図 11 に示すような SD モデルが構築できる。

以上の作業で、因果ループ図からの SD モデル構築作業が完了する。このプロセスをまとめると次の 7 つのステップとなる。

ステップ 1 考察対象のシステムの中に含まれるキーワードから原因と結果の因果関係を抽出する（図 4）。

ステップ 2 抽出した因果関係をさらなる因果関係として関連づけ、因果ループを見いだす（図 5（左））。

ステップ 3 因果ループを発見したら、その因果ループが正のフィードバックループであるのか負のフィードバックループであるのかを特定する（図 5（右））。

ステップ 4 因果ループに含まれるストックを 1 つ以上見いだし、そのキーワードを四角形で囲む（図 6）。

ステップ 5 ストックが隣接する場合には、フローとなる変数を間に追加する（図 10（右））。

ステップ 6 ストックに入ってくる矢印が正の因果関係であればそのストックに対するインフローに置換し（図 7（左））、負の因果関係であればそのストックからのアウトフローに置換する（図 7（右））。

ステップ 7 このようにして出来上がったＳＤモデルのフレームワークに数値データを入れてＳＤモデルを完成させ（図 8）、シミュレーションを実行する（図 9）。

3　システム思考８基本型

以上が私たちが提案する因果ループから SD モデルを効率よく構築する方法論である。この方法論の有効性を検証するために、システム思考８基本型を取り上げ、SD モデルの構築を試みることにする[6]。システム思考基本型 (Systems Archetypes) は、センゲの著書 [5] の Appendix 2 (pp. 378 - 390) の中で最初に取り上げあられた。同書の邦訳「最強組織の法則」では、残念ながらこの部分が割愛され日本の読者の目に触れることはなかった。その後、Kim and Anderson[2] によって体系的に整理され、システム思考８基本型として経営コンサルティングの現場や学校教育の中で広く利用されるようになった。

[6] ８基本型をＳＤモデルに展開した先行研究 [1] に於いても基本型の因果ループからＳＤモデルを構築するプロセスそのものに対して精査していない。

システム基本型とは、組織や人間活動等が作り出す様々な振る舞い (Behaviors) の中で、特定のパターン運動が繰り返し起こることに着目し、それらをもたらすシステムの構造は、そうした振る舞いの如何を問わずすべて同型のものとしてとらえられるという見方に立脚している。こうしたシステムの基本構造を知ることで、複雑化した問題を単純化して捉えることが出来るようになり、様々な振る舞いをもたらすシステム構造の型を統一的に把握出来るようになる。

システム思考基本型として以下の８つがある。

- 応急処置の失敗 (Fixes That Fail)
- 問題の転嫁 (Shifting the Burden)
- 成功の限界 (Limits to Success)
- 目標のなし崩し (Drifting Goals)
- 成長と投資不足 (Growth and Underinvestment)
- 成功には成功を (Success to the Successful)
- エスカレート (Escalation)
- 共有地の悲劇 (Tragedy of the Commons)

こうした基本型を学習する利点は、こうした基本型があたかもビルディングブロックのようにして複雑なシステムの部分を構成している、換言すればこうした基本型を組み合わせることによって、複雑なシステムの構築が容易になるということである。MIT のジョン・スターマン教授が主張するように、システム基本型に頼ることなく、より自由度の多い因果ループ図やストック・フロー図を駆使しながらモデル表現をすることは重要である。事実、彼の代表的 SD テキスト [7] は、こうした基本型について全く触れられていない。また、筆者の教育実践の経験からしても、こうした基本型の学習の結果、これらを組み合わせればモデルが構築できるといった観念にとらわれ、自由な発想によるモデリングが阻害されることになるといった弊害も考えられる。

にもかかわらず、私たちがシステム基本型にこだわるのは、東洋の知恵が結集されたことわざをシステム基本型のような因果ループで表現することの有用性である。東洋思想の根底に横たわる無常の世界観と因果関係、因果ループでシステム構造を解明するシステム思考には、親密性が存在する[7]。「無常」とは、常態ではあり得ず、ダイナミックに動き回るという世界観であり、まさにシステムのダイナミズムそのものである。即ち、因果ループから SD モ

[7]余談であるが、システム思考基本型の提唱者であるセンゲは、サンフランシスコから金門橋 (Golden Bridge) を渡った山間部にあるグリンガルチ禅センターを訪れ、座禅の修行をしたことがあると、ある国際 SD 学会で筆者の一人に語ってくれた。この禅センターは筆者の一人がカリフォルニア留学時代に頻繁に座禅に通った禅堂でもあり、彼の探求心に感銘を受けた。

デルを構築するという本論文の主旨は、東洋思想の科学的分析の序説ともなり得るのである。

以下、こうした観点からシステム思考の８基本型を順次考察してゆく。

4　応急処置の失敗

4.1　ことわざ

経験則に基づいた類似の事例が、パターン化して知恵となり、それが「ことわざ」となって社会に根付いている。システム思考によると、こうしたことわざが繰り返し語り継がれるのは、そうしたことわざを生み出すシステム構造が社会に存在しているということになる。よって、私たちの古来からのことわざとそれを生み出すシステムの（因果ループ）構造を関連づける試みは、非常に有用である。経営者は、多くのことわざを座右の知恵、経営の指針とし、また実際の経営戦略にも活用している。ことわざとシステムの構造の関係を考察することは、これまでになかった新しい角度から経営における行動パターンをシステム構造的に分析することになる。

以下では、様々なシステムの構造が繰り返し生み出す、８つの代表的振る舞い・パターンに該当することわざを探り出してゆくことにする。まずは、以下のことわざを観察してみよう[8]。

- 一難去ってまた一難
- 悪事千里を走る
- 悪事、身にかえる
- 身から出た錆
- 自業自得
- 過ぎたるは及ばざるがごとし
- 薬も過ぎれば毒となる

これらに共通している知恵とは何だろうか。どんな共通のパターンが読み取れるであろうか。「無常」な世界に生きている私たちは、また私たちの組織は、日々いろいろな問題に直面し、そうした問題点に対処し、解決策を見いだしながら日々活動している。問題は、その解決方法である。善意、悪意に関わらず、もし解決策が応急処置的で根本的な問題解決につながらなければ、いずれそのツケが回ってくるよと、こうしたことわざは警告しているのではないだろうか。コスト増という問題に直面し、賞味期限の切れた商品のラベ

[8]以後本論文で紹介することわざは、筆者の一人の講義ノート [11] 及び [12] に負っている。MBA クラスの受講生から教えられたものも多々あり、彼らの英知にも感謝！。

ルを張り替えて再出荷したり、客が残した食材を使い回す等々といった応急処置をとった企業の末路を想起すれば十分である。

4.2 因果ループ図

こうしたことわざにピッタリなのが「応急処置の失敗」の基本型である。応急処置の失敗のシステム構造は次のとおりである。

まず「問題」に対して短期的視点での対処（即ち「応急処置」）をすることにより、「問題」が軽減する。しかしながら、時間の遅れを伴って[9]、「意図しない結果」が発生し、「問題」を悪化させるというものである。「応急処置の失敗」の因果ループを図 12 に示す。

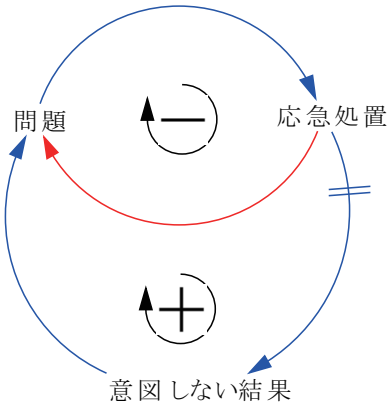

図 12: 「応急処置の失敗」の因果ループ図

次に、SD モデルを構築するためにストック変数を見つけよう。そこで時間を停止してみと、その時点で存在しうるのは、「問題」と「意図しない結果」となる。応急処置は、時間を停止すれば応急処置もなされないので消滅する。ストックを明示した因果ループ図を図 13（左）に示す。

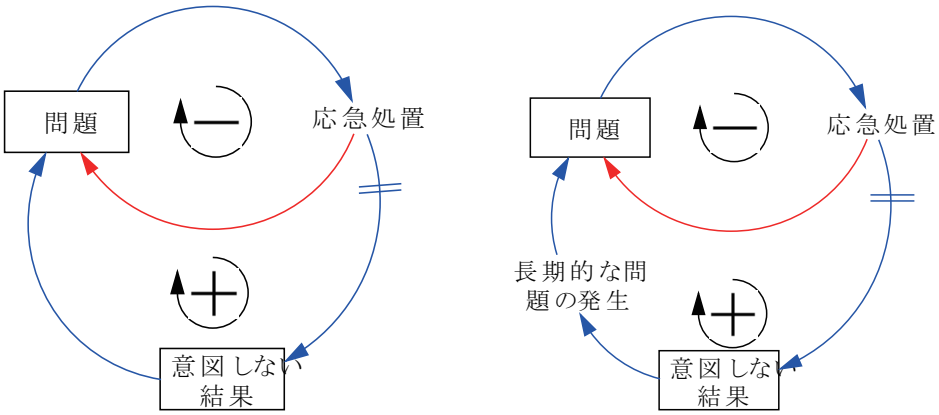

図 13: ストック明示の因果ループ図

ここで、問題と意図しない結果のストックは互いに隣接することになるので、ステップ 5 に従い、その間に長期的な問題の発生というインフローを挿入する。結果を図 13（右）に示す。

[9]時間の遅れを、矢印に 2 本線を引いて表している。

4.3　SD モデル

SD モデルを構築するために、図 13（右）に対してステップ 6 を、ストック「問題」と「意図しない結果」に適用する。[問題] については、長期的な問題の発生が正の因果関係となり、応急処置の失敗が負の因果関係となっている。[意図しない結果」については、意図しない変化が正の因果関係となっている。よって、図 14 のような SD モデルが構築できる。

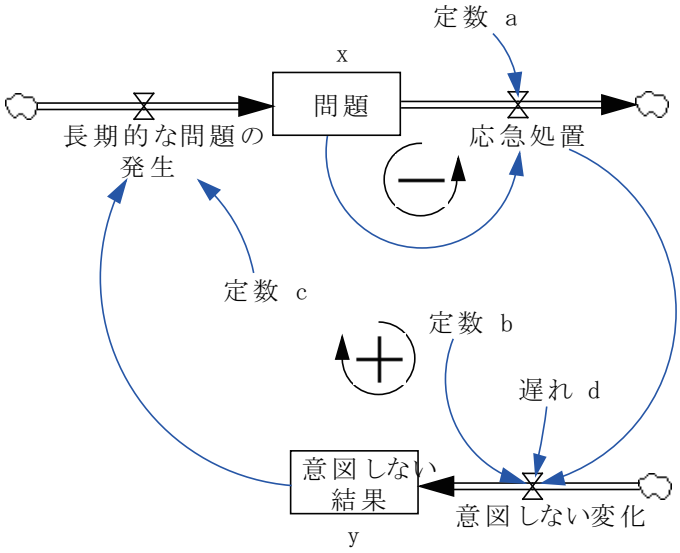

図 14: 「応急処置の失敗」の SD モデル

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE によるプログラミングコードを表 2 に併記する。

表 2: 応急処置の失敗　SD モデル式

問題= INTEG (長期的な問題の発生-応急処置,100)
定数 a = 0.5
定数 b = 0.5
定数 c = 0.5
応急処置= 定数 a * 問題
意図しない変化= DELAY FIXED (定数 b * 応急処置, 遅れ d, 0)
意図しない結果= INTEG (意図しない変化, 0)
遅れ d= 5
長期的な問題の発生= 定数 c * 意図しない結果

4.4　微分方程式系

図 14 の SD モデルにおいて、ストック「問題」を x、「意図しない結果」を y とおき、変数 $x(t), y(t)$ の時間微分を慣例に従ってそれぞれ $\frac{dx(t)}{dt} = \dot{x}, \frac{dy(t)}{dt} = \dot{y}$ で表す。SD モデルにおけるストック変数のネットフロー（＝インフロー－アウトフロー）は、定義によりその時間微分に等しいので次式が得られる[10]。

$$\dot{x} = -ax(t) + cy(t) \tag{1}$$

$$\dot{y} = abx(t - d) \tag{2}$$

初期条件 $x(0) = x_0, y(0) = y_0$.

4.5　シミュレーション

図 15 にシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 100,\ y_0 = 0,\ a = 0.5,\ b = 0.5,\ c = 0.5,\ d = 5$. Time = 0 から 4

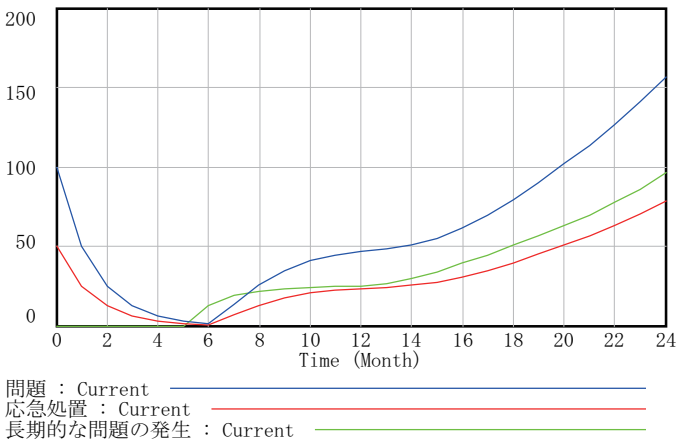

図 15:「応急処置の失敗」のシミュレーション結果

までは、「応急処置」による平衡ループが優勢に作用し、「問題」は減少する。「応急処置」は、時間的遅れを伴って「意図しない変化」をもたらす。Time = 5 以降、「意図しない変化」がインフローとなって発生し、ストック「意図しない結果」が蓄積する。「意図しない結果」は、「長期的な問題の発生」に繋がり、それがインフローとなり「問題」を増加させる。その結果、「長期的な問題の発生」が「応急処置」を上回ることになり、「問題」は上昇に転じ、増強ループ作用により「問題」はますます大きくなってゆく。

[10]本論文の微分方程式系における数式の多くは、Bourguet-Diaz and Perez-Salazar [1] に負っている。

5　問題の転嫁

5.1　ことわざ

読者の中には、ストレスをアルコールで紛らわすようなことがないだろうか。仕事がたまってきたので、それを紛らわすためにアルコールを飲み始めるが、その結果健康を害して、ますます仕事がたまっていってしまう（こんなバズでは！）というのが問題の転嫁の好例である。そうなるのがわかっておれば、最初からアルコールに逃げずに、仕事に立ち向かって頑張ったのにと後悔しても後の祭りである。

また、社内で発生した問題を、外部介入や外部仲裁に頼って解決するようなことはないだろうか。その結果、内部で解決しようという意志がなくなり（自浄効果喪失）、ますます外部依存症となり、内部の人材能力開発のチャンスを逃すのである。

以下のようなことわざが、こうした問題の転嫁に該当すると考えられる。

- 急がば回れ
- 楽あれば苦あり、苦あれば楽あり
- 薬より養生
- 無理は禁物
- 喉元過ぎれば熱さを忘れる
- 木を見て森を見ず
  You cannot see the woods for the trees.
- 死馬の骨を買う

5.2　因果ループ

「問題の転嫁」のシステム構造は次のとおりである。「問題」に直面した場合、２つの異なった解決策が存在するとしよう。一つ目は短期的視点で「応急処置」的に「問題」を軽減させるというものであり、二つ目は、時間的な遅れを伴うがより長期的視点で「根本的解決」を目ざして「問題」を軽減させるというものである。しかしながら、もし「応急処置」的解決策を選択すれば、やがて「意図しない結果」を生み、「根本的解決」を阻害することになる。前述の「応急処置の失敗」は、「応急処置」の副作用として「問題」を悪化させるだけであったが、ここでは問題の根本的解決を図る解決策が新たに加わり、応急処置による副作用がそうした問題の根本的解決を阻害するようになる。「問題の転嫁」の因果ルー図を図 16（左）に示す。

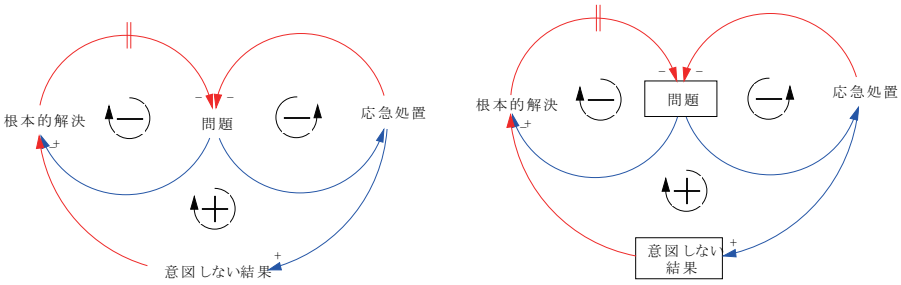

図 16: 「問題の転嫁」の因果ループ図（左）とそのストック明示（右）

次に、SD モデルを構築するためにストック変数を見つけよう。因果ループで表現された変数のうち、時間を停止しても問題は依然として存在し続けるし、また、意図しない結果も存在し続ける。よってストックは、「問題」と「意図しない結果」となる。しかしながら、応急処置や根本的解決は、時間を停止すれば処置は実行されず消滅する。ストックを明示した因果ループ図を図 16（右）に示す。

5.3 ＳＤモデル

SD モデルを構築するために、図 16（右）に対して、第 2 節で説明した変換を加えていく。ステップ 6 を、ストック「問題」と「意図しない結果」に適用すると、図 17 のような SD モデルが出来上がる。

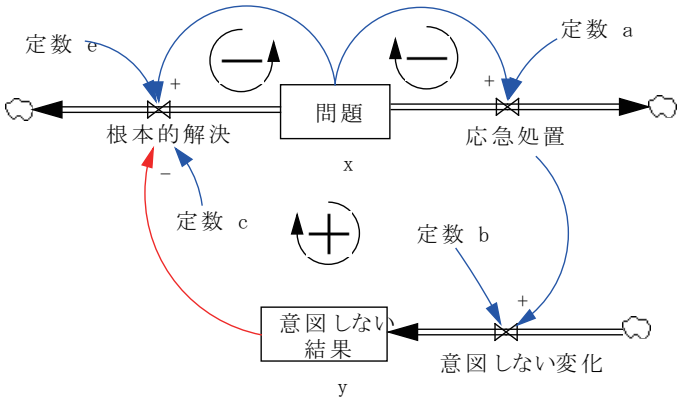

図 17: 「問題の転嫁」の SD モデル

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE による プログラミングコードを表 3 に併記する。

表 3: 問題の転嫁　SD モデル式

問題= INTEG (-応急処置-根本的解決, 100)
定数 a= 0.2
定数 b= 0.5
定数 c= 0.5
定数 e= 0.2
応急処置=定数 a * 問題
意図しない変化= 定数 b * 応急処置
意図しない結果= INTEG (意図しない変化, 0)
根本的解決= 定数 e * 問題-定数 c*意図しない結果

5.4　微分方程式系

図 17 の SD モデルにおいて、「問題」を x、「意図しない結果」を y とおくと、次式が得られる。

$$\begin{aligned} \dot{x} &= -ax(t) - [ex(t) - cy(t)] \\ &= -(a+e)x(t) + cy(t) \end{aligned} \tag{3}$$

$$\dot{y} = abx(t) \tag{4}$$

初期条件 $x(0) = x_0, y(0) = y_0$.

5.5　シミュレーション

図１８にシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 100,\ y_0 = 0,\ a = 0.2,\ b = 0.5,\ c = 0.5,\ e = 0.2.$ Time $= 0$ から 5 付近 までは、「応急処置」による平衡ループと「根本的解決」による平衡ループが優勢に作用し、「問題」は減少する。「応急処置」は、「意図しない変化」をもたらす。「意図しない変化」がインフローとなり、ストック「意図しない結果」が蓄積する。「意図しない結果」は、「根本的解決」を減少させるとともに、さらには「根本的解決」のフローを逆転させ、「問題」を逆に増加させる作用に転じる。結果として、「問題」は上昇に転じ、増強ループにより「問題」は大きくなっていく。

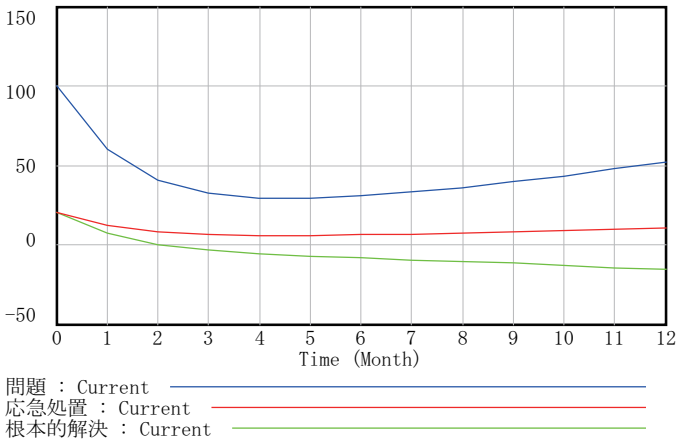

図 18: 「問題の転嫁」のシミュレーション結果

6 成長の限界

6.1 ことわざ

成長の限界とは、文字どおり無限の成長は不可能であるということである。顧客満足 (Customer Satisfaction) が 減少し始める例を考えてみよう。研究開発投資の結果、新製品の販売が増加し、急成長するが、その拡大過程で経営能力が追いつかなくなり、顧客サービスの質が低下し始め、次第に顧客の不満が高まり、顧客満足が減少し始めることになるといったケースである。

事業拡大が行き詰まる他の例として、事業が順調に拡大して売上げが増加し、それがさらなる事業拡大に繋がってゆくといった指数的成長を遂げていたが、市場規模の制約のために次第に市場が飽和状態になり、やがて売上げが伸び悩んでくるといったケースが考えられる。

こうした成長の限界に該当することわざとして、以下のようなものがある。

- おごる平家は久しからず
- 勝って兜の緒を締めよ
- 水泡に帰す
- 好事魔多し
- 自慢は知恵の行き止まり
  Who commends himself berates himself.

こうした「成長の限界」のシステム構造を少し掘り下げて考えてみると、以下の 3 つのケースが想定される。

（1）「成果」の指数的増加をもたらす生成メカニズムと、「成果」の指数的減少をもたらす消滅メカニズムが存在する。初期にはもっぱら生成メカニズムが作用し、「限界」に到達する末期には生成メカニズムと同等の規模で消滅メカニズムが作用して均衡する場合である。例えば、人口が増加し、次第に環境が許容する最大人口に近づき、出生と死亡が均衡して人口が一定になるようなケースである。

（2）「成果」の生成メカニズムのみが存在する。成果の変化の割合は、初期はもっぱら指数的に増加するものの、成果が限界に到達する末期には、成長余地の減少に従って生成がゼロとなるケースである。例えば、一定の人口の島で感染症に罹患してゆく場合である。当初は爆発的に患者が増加するものの、島民全員が罹患してしまう末期では変化の割合である新規に罹患する患者はゼロとなる。（1）は、成長の限界に到達した時点でインフローとアウトフローが均衡するのに対して、（2）は成長の限界に達した時点では、フローそのものがゼロになっている点で異なる。

（3）ストック総和が一定の「減少ストック」と「増加ストック」が存在し、減少ストックから増加ストックへとフローが変動するケースである。例えば、マーケティング市場である商品が潜在需要者（減少ストック）の間に広がってゆき、次第に利用者（増加ストック）が増加してゆくようなケースである。

以下、（1）、（2）、（3）を順次、成長の限界（1）（イン・アウトフローが均衡する場合)、成長の限界（2）（インフローがゼロに収束する場合)、成長の限界（3）（増減ストック総和が一定の場合）に場合分けして順次考察する。

6.2 成長の限界（1）

6.2.1 因果ループ

インフローとアウトフローが均衡する場合に於ける「成長の限界（1）」のシステム構造は次のとおりである。成長がさらなる成長を生成する正のフィードバックメカニズムが存在し、成果は指数的に増加してゆく。こうした過程で、増大した成果がシステムの限界の制約に近づき、やがて成果を消滅させるメカニズムが機能し始める。すなわち、この消滅メカニズムは限界に対して成果の量が小さいときの作用は小さいが、限界に近づけば生成と同規模の成果を逆に消滅させるというものである。「成長の限界（1）」の因果ループを図 19（左）に示す。

次に SD モデル構築のために、ストックを見出そう。因果ループ図で表現された変数のうち成果のみがストックとなる。なぜなら時間を止めても成果は依然として存在し続けるが、生成も消滅も時間を停止すれば作用しなくなるからである。ストックを図示した因果ループ図を図 19（右）に示す。

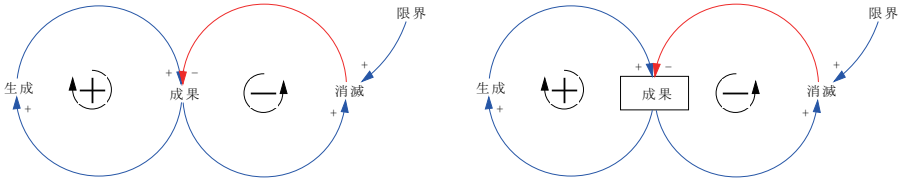

図 19: 「成長の限界（１）」の因果ループ図（左）とそのストック 明示（右）

6.2.2 SD モデル

SD モデルを構築するために、図 19 に対して、第 2 節で説明した変換を加えていく。ステップ 6 を、ストック「成果」に適用すると、図 20 の結果となる。

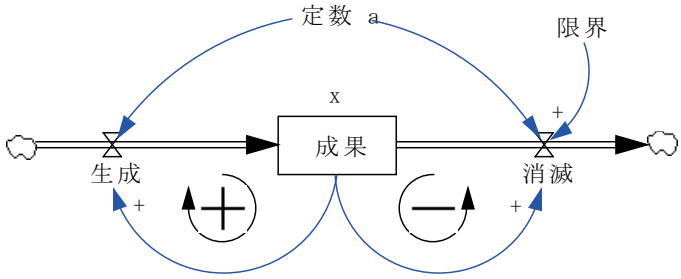

図 20: 「成長の限界（１）」の SD モデル

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE による プログラミングコードを表 4 に併記する。

表 4: 成長の限界（１）SD モデル式

定数 a = 0.1
成果 = INTEG (生成 - 消滅, 1)
消滅 = 定数 a * 成果*(1-(限界-成果)/限界)
生成 = 定数 a * 成果
限界 = 100

6.2.3 微分方程式系

図 20 の SD モデルにおいて、「成果」を x、限界を L とおくと、次式が得られる。

$$\dot{x} = ax(t) - ax(t)\left(1 - \frac{L - x(t)}{L}\right)$$
$$= a\left(1 - \frac{x(t)}{L}\right)x(t) \tag{5}$$

初期条件: $x(0) = x_0$.

6.2.4 シミュレーション

図 21 に SD モデルのシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 1, a = 0.1, L = 100$. Time = 0 から Time = 50 付

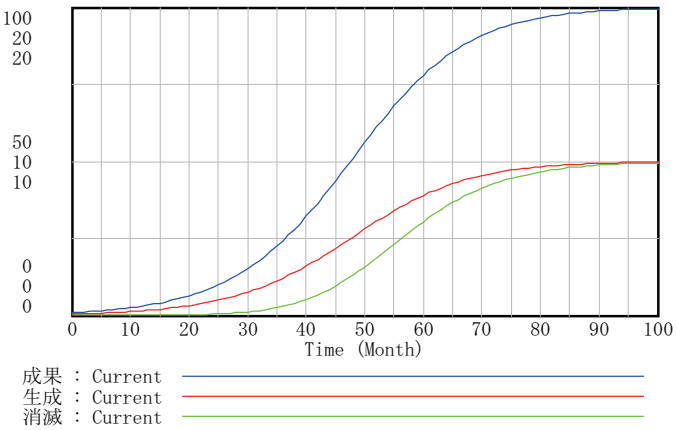

図 21: 「成長の限界（1）」のシミュレーション結果

近までは、もっぱら生成が優勢であり「成果」が加速的に蓄積する。Time = 5 0 付近では、生成に対して約５０%の割合で消滅が発生するようになる。結果的に、徐々に「成果」の蓄積のスピードが低下する。Time = 100 付近では、生成に対して、同量の消滅が発生し、「成果」への正味のインフローは無くなり、「成果」は限界に達する。

6.3 成長の限界（2）

6.3.1 因果ループ

インフローがゼロに収束する場合に於ける「成長の限界（2）」のシステム構造は次のとおりである。成果を更に生成するメカニズムが存在する。すなわち、成果と成長ギャップの積に比例して成果の生成量がコントロールされるというメカニズムである。成果の量が限界に比べて圧倒的に小さい初期で

は成長のギャップが大きく、生成のフィードバックが増強的となり、成果の量は指数的に増加する。しかし、成果の量が限界に近づく末期では成長ギャップは小さくなり、成果の生成は止まる。「成長の限界（２）」の因果ループを図 22（左）に示す。

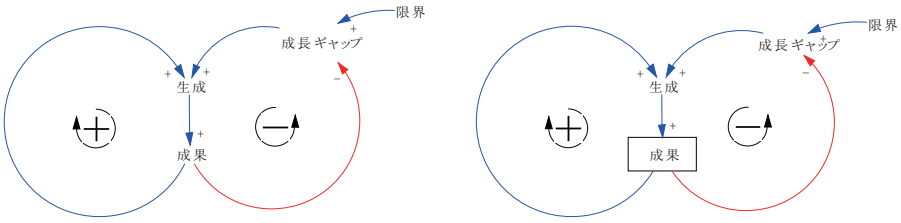

図 22: 「成長の限界（２）」の因果ループ図（左）とそのストック 明示（右）

次に SD モデル構築のために、ストックを見出そう。明らかに因果ループで表現された変数のうち成果がストックとなる。ストックを図示した因果ループ図を図 22（右）に示す。

6.3.2 SD モデル

SD モデルを作成するために、図 22 に対して、第２節で説明した変換を加えていく。ステップ６を、ストック「成果」に適用すると、図 23 の結果となる。

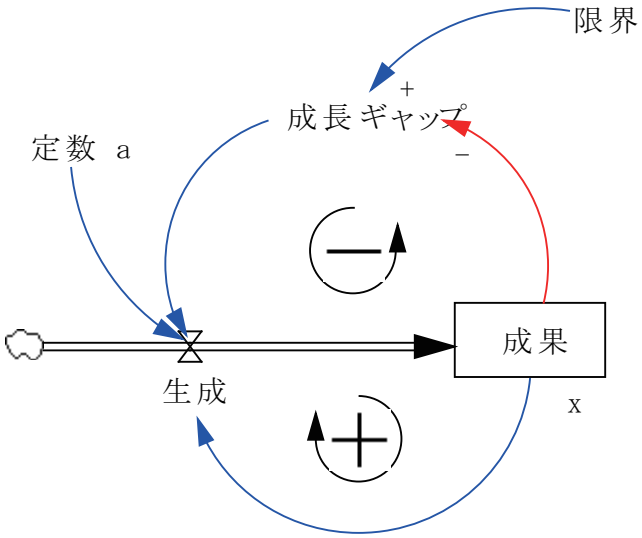

図 23: 「成長の限界（２）」の SD モデル

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定

数を充足して数値データを入れる。VensimPLE によるプログラミングコードを表 5 に併記する。

表 5: 成長の限界（２）SD モデル式

定数 a = 0.1
成果= INTEG (生成, 1)
成長ギャップ = (限界 - 成果)/限界
生成= 定数 a * 成長ギャップ * 成果
限界=100

6.3.3 微分方程式系

図 23 の SD モデルにおいて、「成果」を x、限界を L とおくと、次式が得られる。

$$\begin{aligned} \dot{x} &= ax(t)\left(\frac{L-x(t)}{L}\right) \\ &= a\left(1-\frac{x(t)}{L}\right)x(t) \end{aligned} \tag{6}$$

初期条件: $x(0) = x_0$.

6.3.4 シミュレーション

図 23 の SD モデルのシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 1, a = 0.1, L = 100$. Time = 0 から Time = 50 付近までは、「成果」が「生成」を生む増強ループが優勢で、急速に「生成」が増加し、「成果」も蓄積する。「成果」が蓄積するに従って、「成果」から成長ギャップを経て生成に至る平衡ループが優勢となる。その結果、生成は Time = 50 付近で極大値をとり[11]、その後徐々に減少し、Time = 100 付近で再びゼロになる。この生成をインフローとして「成果」は Time = 0 付近のゼロから加速的に増加し、Time = 50 で加速のピークを迎え、Time = 100 付近で成長の限界を迎える。

[11] この点のことを変曲点 (Inflection Point) という。

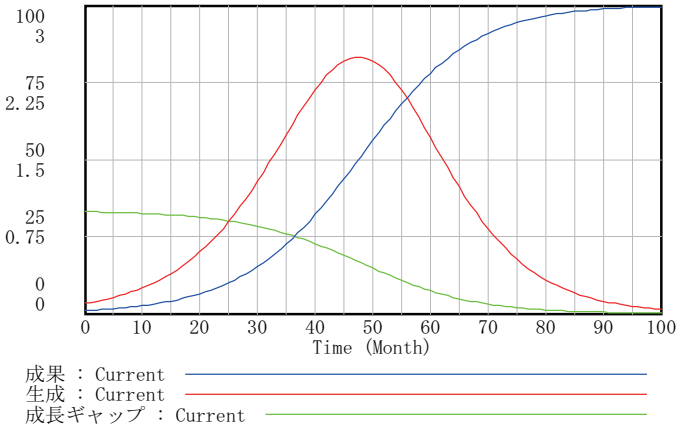

図 24: 「成長の限界（２）」のシミュレーション結果

6.4 成長の限界（３）

6.4.1 因果ループ

増減ストック総和が一定の場合に於ける「成長の限界（３）」のシステム構造は次のとおりである。成果と成果の余地の和が一定という関係に着目する。例えばマーケティング市場におけるある商品の利用者（成果）と潜在的利用者（成果源）のような関係が該当する。成長の限界は、成果と成果源の総和として暗黙的にモデルの初期条件として組み込まれている。ここでは、成果の量に比例して変化させ成果を増加させようとするメカニズムと、成果源に比例して変化させ成果を抑制しようとするメカニズムが存在する。そして、成果と成果源の間には常に総和が一定となるという条件が存在する。成長の限界（３）の因果ループを図 25（左）に示す。

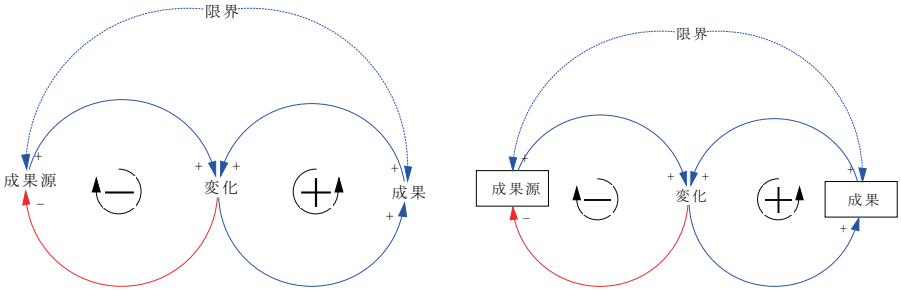

図 25: 「成長の限界（３）」の因果ループ図（左）とそのストック 明示（右）

次に SD モデル構築のために、ストックを見出そう。因果ループで表現された変数のうちストックは、「成果」と「成果源」となる。なぜならば、時間を止めても、成果と成果源は依然として存在し続けるからである。しかしな

がら、変化は、時間を停止すれば当然に消滅する。ストックを図示した因果ループ図を図 25（右）に示す。

6.4.2 SD モデル

SD モデルを作成するために、図 25 に対して、第 2 節で説明した変換を加えていく。ステップ 6 を、ストック「成果」に適用すると、図 26 の結果となる。

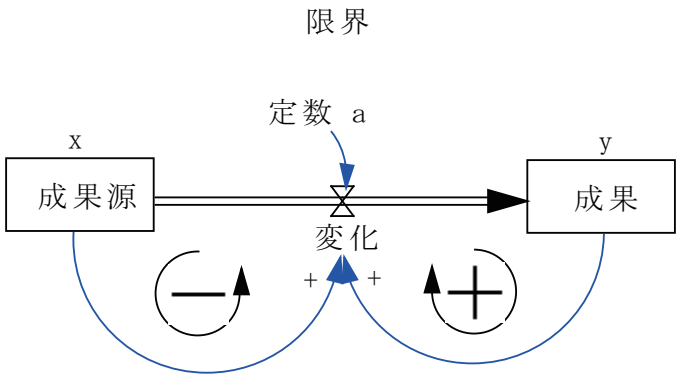

図 26: 「成長の限界（3）」の SD モデル

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE による プログラミングコードを表 6 に併記する。

表 6: 成長の限界（3）SD モデル式

定数 a = 0.001
変化 = 定数 a * 成果 * 成果源
成果 = INTEG (変化, 1)
成果源 = INTEG (-変化, 99)
限界=100

6.4.3 微分方程式系

図 26 の SD モデルにおいて、「成果」を y、「成果源」を x とおき、限界を $L = x + y$ と定義すると次式が得られる。

$$\begin{aligned}\dot{y} &= ax(t)y(t) \\ &= a(L - y(t))y(t) \end{aligned} \tag{7}$$

初期条件: $x(0) = x_0$

6.4.4 シミュレーション

図 27 に SD モデルのシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$L = 100, a = 0.001, x_0 = 1$ Time $= 0$ において、「成果」

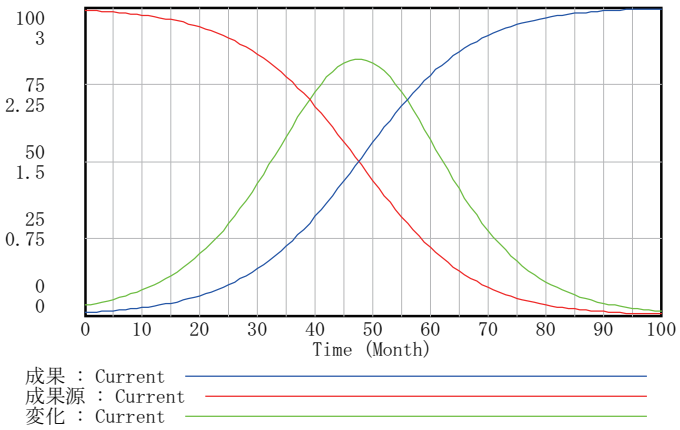

図 27: 「成長の限界（3）」のシミュレーション結果

はゼロ、「成果源」は「成果」の成長の限界値をとっている。2 つのストックの間のフローである「変化」は、Time $= 0$ でゼロ、Time $= 50$ で極大値となり、Time $= 100$ で再びゼロになる。それに従ってストック「成果」は Time $= 0$ のゼロから加速的に増加し、Time $= 50$ で加速のピーク（変曲点）を迎え、Time $= 100$ 付近で成長の限界を迎える。

7 目標のなし崩し

7.1 ことわざ

目標のなし崩しの例として、製品の引き渡し時間を延長するケースを考えよう。製品の引き渡し時間（目標）が守られなくなれば、生産計画を見直して引き渡し時間の短縮をはかるように対処すべきであるのに、そうせずに、逆に引き渡し時間の延長をはかるといった場合である。

目標のなし崩しに該当することわざとして、以下のようなものが考えられる。

- 為せば成る為さねば成らぬ何事も
  （成らぬは人の為さぬなりけり： 米沢藩主上杉鷹山）
- 歳寒（さいかん）) の松柏（しょうはく)、松柏の操
- 精神一到何事か成さざらん
- 無い袖は振れぬ
- 同床異夢
  The horse thinks one thing, and the person that saddles it another.
- 問うに落ちず語るに落ちる
  The tongue is ever t n urning to the aching tooth.

7.2 因果ループ

目標のなし崩しのシステム構造は次のとおりである。目標と実現値の差である達成ギャップが生じた場合、それを埋めるために 2 つの異なった行動が存在する。一つ目は改善活動により実現値を大きくし達成ギャップを埋める行動である。二つ目は目標をなし崩し的に修正して達成ギャップを埋める行動である。「目標のなし崩し」の因果ループ図を図 28（左）に示す。

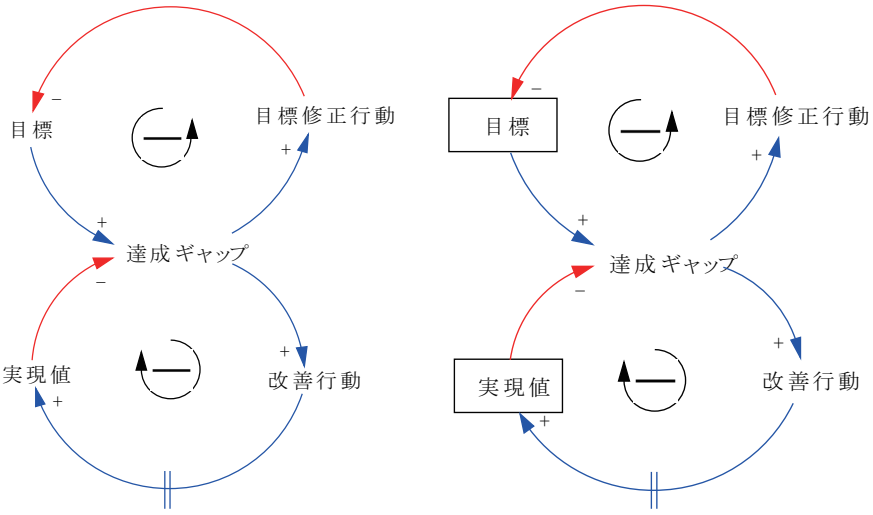

図 28: 「目標のなし崩し」の因果ループ図（左）とそのストック 明示（右）

次に SD モデル構築のために、ストックを見出そう。ここでは、「目標」と「実現値」の 2 種類のストックが存在する。目標は一見すると、上司の意思によって決定すべきパラメータのように見える。しかしながら、ここで考える目標とは、上司の一存で決めるべきものではなく、現有の組織能力の精査や

ステークホルダーズの理解が得られるかの吟味を伴って設定されるものであり、組織の真の実力値の見極めやステークホルダーズとのコミュニケーション能力の結果、設定を変更しうるという意味で、一種の組織能力の反映としての目標であると考える。それゆえ、「目標」は単なる情報（パラメータ）ではなくストックと考えるのが相当である。「実現値」についても、かかるパフォーマンスを表出しうる組織能力の反映であるから、ストックである。実現値の向上行動が結果に反映されるストックを明示した因果ループ図を図 28（右）に示す。

7.3 SD モデル

SD モデルを構築するために、図 28 に対して、第 2 節で説明した変換を加えていく。ステップ 6 を、ストック「目標」、「実現値」に適用すると、図 29 の結果となる。

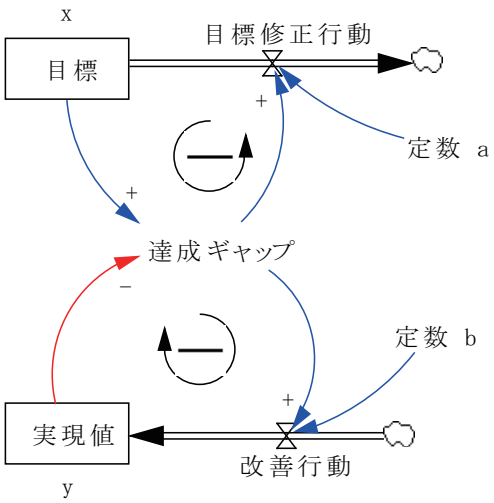

図 29: 「目標のなし崩し」の SD モデル

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE によるプログラミングコードを表 7 に併記する。

7.4 微分方程式系

図 29 の SD モデルにおいて、ストック「目標」を x、ストック「実現値」を y とおくと、次式が得られる。

$$\dot{x} = -\frac{x(t) - y(t)}{a} \tag{8}$$

表 7: 目標のなし崩し　SD モデル式

定数 a = 12
定数 b = 24
達成ギャップ = 目標 - 実現値
実現値 = INTEG (改善行動, 50)
改善行動 = 達成ギャップ / 定数 b
目標= INTEG (- 目標修正行動, 100)
目標修正行動 = 達成ギャップ / 定数 a

$$\dot{y} = -\frac{x(t) - y(t)}{b} \tag{9}$$

初期条件: $x(0) = x_0, y(0) = y_0$.

7.5　シミュレーション

図 30 に SD モデルのシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 100, y_0 = 50, a = 12, b = 24$.

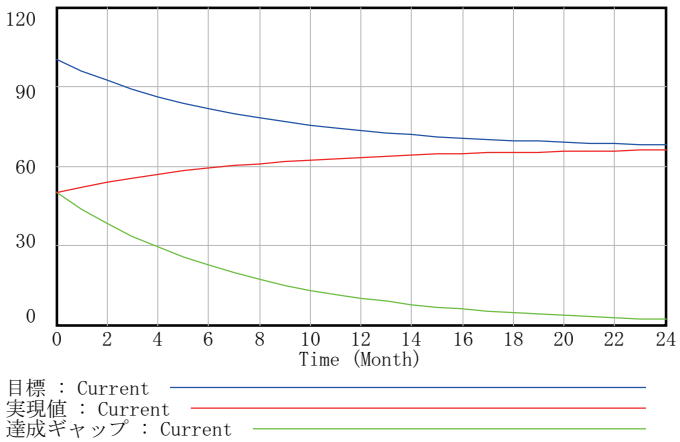

図 30: 「目標のなし崩し」のシミュレーション結果

Time = 0 において、「目標」は 100、「実現値」は 50 で、達成ギャップは 50 である。この達成ギャップを埋めるため、「実現値」は「改善行動」によって徐々に上昇する。また、「目標」も「目標の修正行動」によって低下する。

これらの２つの動きによって、「目標」と「実現値」の達成ギャップは徐々に小さくなってゆく。なお、このシミュレーションでは、改善行動のための時間調整定数 b に対して、目標修正行動の時間調整定数 a を小さく設定しているので、改善行動による「実現値」の改善よりも目標修正行動による「目標」のなし崩しの方がより多く発生している。

8　成長と投資不足

8.1　ことわざ

この基本型でよく引き合いに出されるのが、People's Express Airline の例である [2]。同社は格安料金で急成長し、旅客数の増加がさらなる収益増をもたらし、それをもとに飛行機数と飛行時間を拡大し、それがさらなる旅客数の増加に繋がるといったように、指数的成長を遂げていたが、旅客数の増加は次第にサービスの低下（オーバーブッキング、運行時間の遅れ等）をもたらしてくるようになり、やがて同社の評判が低下し、旅客数の増加も限界になってくるといった事例である。こうした状況を打破するために、同社は、顧客サービスの向上のために投資し、サービス能力を増大すべきであった。

成長と投資不足に該当することわざとして、以下のようなものが考えられる。

- 備えあれば憂いなし
- 霜を履んで堅氷至る
- 取らぬ狸の皮算用
- 郷に入りては郷に従え
- 泥棒を捕らえて（見て）縄を綯う

8.2　因果ループ図

成長と投資不足のシステム構造は次のとおりである。成果と生成の正のフィードバックループ存在し、成長が加速するが、やがて、生産能力の制約を受けてパフォーマンスが低下し、成果を抑制するようになる。ここまでは「成長の限界」の構造と同じである。但し、限界は不変ではなくて、能力開発への投資という形で生産能力をアップできるという点が新たに加わる。しかしながら、一般的に能力開発への投資は長期に及ぶので手遅れになる可能性が高い。結果的に、成果は飽和するかあるいは、十分な成長の割合を確保できないこととなる。成長と投資不足の因果ループ図を図 31（左）に示す。

次に SD モデル構築のために、ストックを見出そう。このシステムにおいて、ストックは「成果」と「生産能力」となる。成果はその背後に生成の増

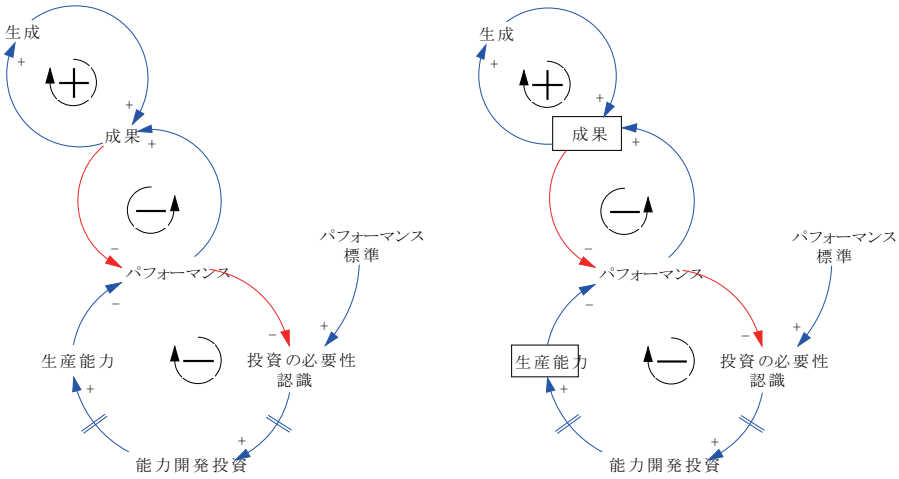

図 31: 「成長と投資不足」の因果ループ図（左）とそのストック 明示（右）

強ループを含んでいる。また、生産能力は投資による能力開発のメカニズムを含んでいる。成果と生産能力の差がパフォーマンスとなる。パフォーマンスは、成果に対しても生産能力に対してもネガティブフィードバックされ、平衡ループの一環として作用する。ストックを図示した因果ループ図を図 31（右）に示す。

8.3 SD モデル

SD モデルを作成するために、図 31（右）に対して、第 2 節で説明した変換を加えていく。ステップ 6 を、ストック「成果」と［生産能力」に適用すると、図 32 の結果となる。

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE によるプログラミングコードを表 8 に併記する。

8.4 微分方程式系

図 32 のＳＤモデルにおいて、成果を x、生産能力を y とおくと次式が得られる。

$$\dot{x} = ax(t)(y(t) - x(t)) \tag{10}$$

$$\dot{y} = c[b - (y(t) - x(t))] \tag{11}$$

初期条件: $x(0) = x_0, y(0) = y_0$.

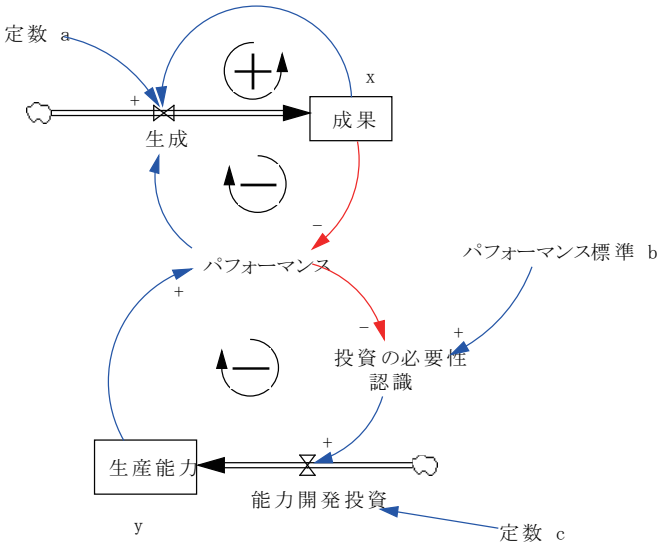

図 32: 「成長と投資不足」の SD モデル

8.5 シミュレーション

図 32 の SD モデルのシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 1, y_0 = 100, a = 0.01, b = 20, c = 15$. 「成果」か

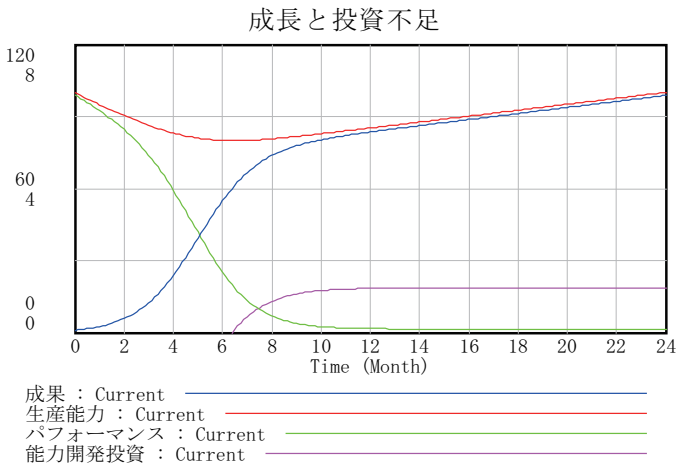

図 33: 「成長と投資不足」のシミュレーション結果

ら生成への増強ループにより、「成果」が指数的に増加する。そして、「成果」が蓄積するに従って、「成果」からパフォーマンスを経て生成に至る平衡ループが優勢となり、パフォーマンスが低下し、「成果」も限界に達する。Time = 6 付近まではパフォーマンスがパフォーマンス標準を上回り、「生産能力」は過剰と判断され、「生産能力」のアウトフローが生じ「生産能力」は低下す

表 8: 成長と投資不足　SD モデル式

定数 a = 0.01
定数 c = 15
パフォーマンス基準 b = 20
成果= INTEG (生成, 1)
生成 = 定数 a * 成果 * パフォーマンス
パフォーマンス = 生産能力 - 成果
投資の必要性認識 = パフォーマンス基準 b - パフォーマンス
生産能力= INTEG (能力開発投資, 100)
能力開発投資 = 投資の必要性認識 / 定数 c

る。Time = 6 付近以降は、パフォーマンスがパフォーマンス標準を下回り、「生産能力」は不足と判断され、「能力開発投資」が開始され、「生産能力」は徐々に上昇する。しかしながらパフォーマンスを回復させ、「成果」を再加速させるまでには至らず、成果は低い生産能力の制約を受けて頭打ちとなる。

9　成功には成功を

9.1　ことわざ

この基本型の例として、昇進の機会を考えよう。The rich becomes richer and the poor becomes poorer のように、偶然のきっかけで昇進の機会が与えられたAさんは、ますます上司から好意的に扱われるようになり、それがさらなる出世の機会となり、やがて出世街道を突っ走りはじめる。一方、ふとしたきっかけで昇進の機会をなくしたBさんは、ますます上司から疎んじられるようになり、やがて昇進の機会から遠ざかるといった不条理な構造のワナにはまってしまう。いわゆる偶然のいたずらが勝ち組と負け組をもたらすシステム構造の一例である。

成功には成功をに該当することわざとして、以下が考えられる。

- 先んずれば人を制す
- 先手は万手
- 機先を制する
- 早い者勝ち
  First come, first served.
- 泣きっ面に蜂

- 弱り目に祟り目
- 傷口に塩
- 惚れた欲目
- 長いものには巻かれろ

9.2 因果ループ

「成功には成功」のシステム構造は次のとおりである。この基本型は一定量の資源を２つの活動が取り合うシステムである。資源の配分は、過去の成功量（信用）に従ってなされる。「成功には成功」の因果ループ図を図 34（左）に示す。

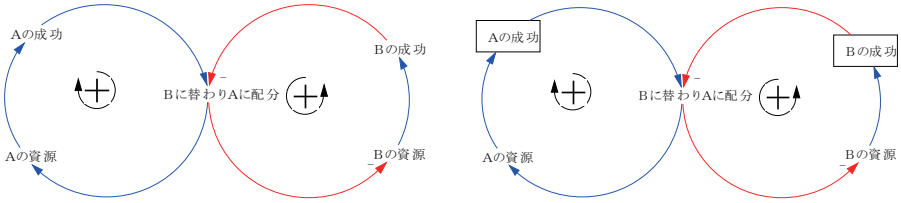

図 34: 「成功には成功」の因果ループ図（左）とそのストック明 示（右）

次に SD モデル構築のために、ストックを見出そう。ストックは、過去からの成功量あるいは成功に基づいて蓄積された信用である。一方フローは、資源の配分の付け替えである。資源は総量が一定であるので、一方の分け前が増加すれば他方は減少する関係にある。図 34（右）は、ストックを図示した因果ループ図である。

9.3 SD モデル

SD モデルを作成するために、図 34（右）に対して、第２節で説明した変換を加えていく。ステップ６を、ストック「Ａの成果」、「Ｂの成果」に適用すると、図 35 の結果となる。

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE によるプログラミングコードを表 9 に併記する。

9.4 微分方程式系

図 35 のＳＤモデルにおいて、Ａの成果を x、Ｂの成果を y とすると次式が得られる。

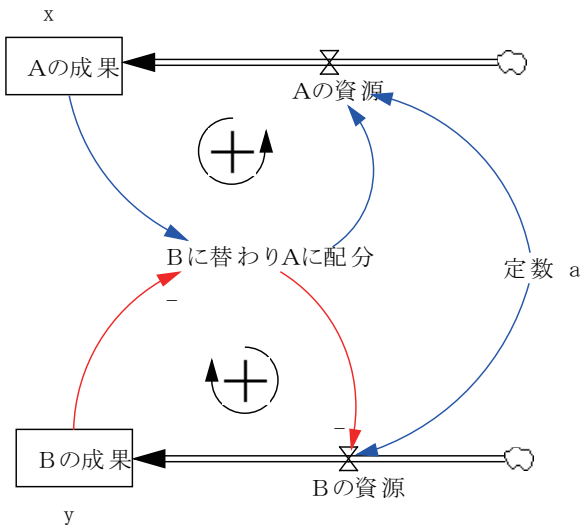

図 35: 「成功には成功」の SD モデル

表 9: 成功には成功　SD モデル式

定数 a = 0.1
Aの成果 = INTEG (Aの資源, 5.5)
Aの資源 = 定数 a * Bに替わりAに配分
Bに替わりAに配分 = Aの成果 - Bの成果
Bの成果 = INTEG (Bの資源, 4.5)
Bの資源 = - 定数 a * Bに替わりAに配分

$$\dot{x} = ax(t) - ay(t) \tag{12}$$

$$\dot{y} = -bx(t) + by(t) \tag{13}$$

初期条件: $x(0) = x_0, y(0) = y_0$.

9.5　シミュレーション

図 36 に SD モデルのシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 5.5, y_0 = 4.5, a = 0.1$. Time = 0 における「Aの成果」は 5.5、「Bの成果」は 4.5 であり、僅かに 1 だけAが有利である。この差によって、BからAへの資源の付け替えが起こり、「Aの成果」はより多

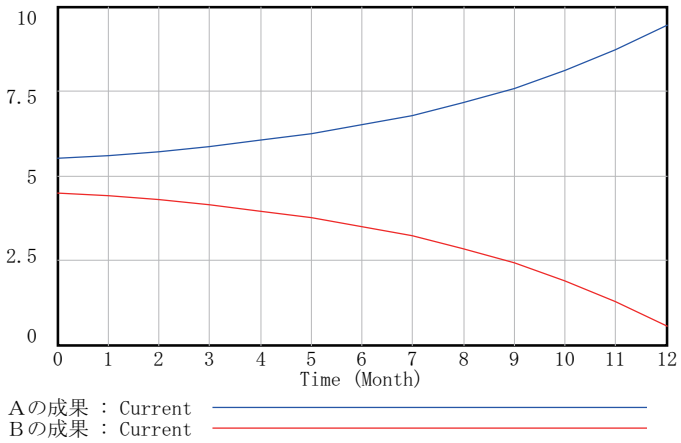

図 36: 「成功には成功」のシミュレーション結果

くなり、「Bの成果」はより小さくなる。この繰り返しによって、「Aの成果」と「Bの成果」のギャップはさらに大きくなっていく。

10 エスカレート

10.1 ことわざ

エスカレートとは、競争が激化することである。例えば、A さんが、B さんとの競争の結果、格差を広げ、こうした格差の拡大がBさんに脅威となる場合を考えよう。そこでBさんは、この脅威を克服するためによりいっそう活発に努力し、やがてよい結果を出し、Aさんとの格差が逆転する。今度は逆にAさんがそれを脅威と感じ、さらに活動を強める。こうして競争はますます激化し、エスカレートする。

価格競争（戦争）の消耗戦も好例である。例えば、A社は価格を引き下げた結果、順調に売り上げを伸ばすことになり、さらに同社の市場シェアも拡大することになる。こうした状況に脅威を感じたB社は、同様に価格切り下げ戦略を採用し、その結果同社の売上げも、市場シェアも回復、拡大することになる。こうしたB社の戦略に脅威を感じたA社は、さらなる価格切り下げに走る。このようにしていったん価格引き下げ競争に巻き込まれると、競争相手とともに、価格下落の正のフィードバックループにはまり込み、こうした価格戦争のワナから抜け出せなくなる。

エスカレートに該当することわざとして、以下のようなのが考えられる。

- 目には目を歯には歯を
  An eye for an eye, and a tooth for a tooth.

- 隣の芝生は青い
- 疑心、暗鬼を生ず
- 血で血を洗う
  Blood will have blood.

10.2　因果ループ

「エスカレート」のシステム構造は次のとおりである。エスカレートは、競争的な状態にある２つの主体の間で、相手に対して一定の関係を保持しようとする。その際に一方が成果を増加させる行動をとることが、他方に脅威となって防御的成果を増加させる行動に走らせることになり、それがまた前者の脅威となるといった競争激化を招くことになる。その結果、いずれもが望まないような高いレベルまで成果を増加させる活動が繰り返される。なお、「成功には成功を」は資源が一定であったが、エスカレートでは相対的立場を一定に保持しようとする。「エスカレート」の因果ループを図 37（左）に示

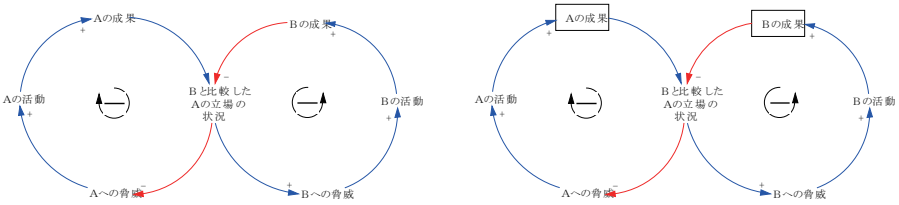

図 37:「エスカレート」の因果ループ図（左）とそのストック明 示（右）

す。この因果ループ図では、２つの平衡ループが描かれているが、ここで重要なのは、Ｂと比較したＡの立場の状況が、相手の活動によって変化し続けていることである。

次に SD モデル構築のために、ストックを見出そう。ここでは明らかに、ストックはＡの成果、Ｂの結果である。Ａの脅威、Ｂの脅威もストックであるとの理解も可能であるが、ここでの脅威は、過去に認識していた脅威からの変化量が重要な意味をもち、その変化量がそれぞれの活動へと突き動かす原動力となるとの理解から、両者の成果の相対関係から導き出される情報として取り扱う。ストックを図示した因果ループ図を図 37（右）に示す。

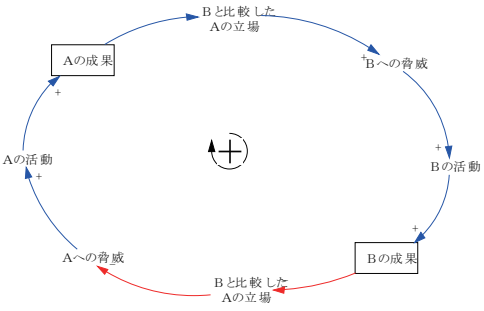

図 38:８字型因果ループへの展開図

このエスカレートの８の字型の因果ループ図を、Untwisted Struture の構造に展開すると、ひとつの正のフィードバックループとなる。競争的状況がエスカレートするのは、いつでもこうした正のフィードバック構造が潜んでいるからである。図 38 に「エスカレート」の８字型因果ループの展開図を示す。

10.3 SD モデル

SD モデルを作成するために、図 37 に対して、第２節で説明した変換を加えていく。ステップ６を、ストック「Aの成果」、「Bの成果」に適用すると、図 39 の結果となる。

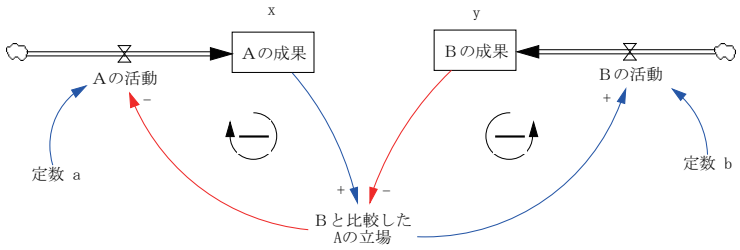

図 39: 「エスカレート」の SD モデル

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE によるプログラミングコードを表 10 に併記する。

表 10: エスカレート SD モデル式

定数 a = 2
定数 b = 1
Aの成果 = INTEG (Aの活動, 40)
Aの活動 = 定数 a - Bと比較したAの立場
Bと比較した A の立場 = Aの成果/Bの成果
Bの成果= INTEG (Bの活動, 10)
Bの活動= - (定数 b - Bと比較したAの立場)

10.4 微分方程式系

図 39 のＳＤモデルにおいて、Aの成果を x、Bの成果を y とおくと次式が得られる。

$$\dot{x} = a - \frac{x(t)}{y(t)} \tag{14}$$

$$\dot{y} = -\left(b - \frac{x(t)}{y(t)}\right) \tag{15}$$

初期条件: $x(0) = x_0, y(0) = y_0$.

10.5 シミュレーション

図 40 にＳＤモデルのシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 40, y_0 = 10, a = 2, b = 1$. Time $= 0$ における「Ａ

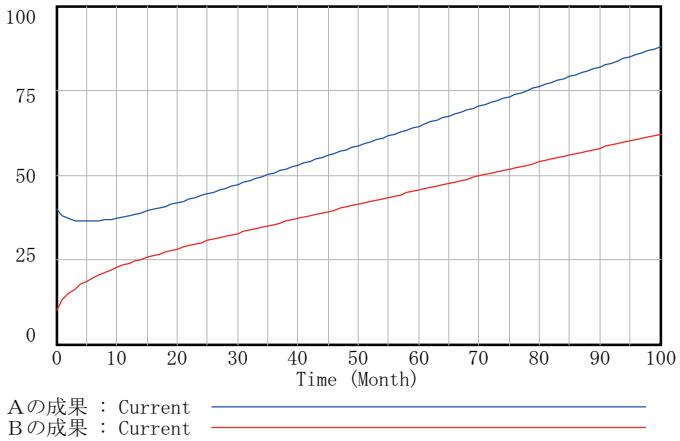

図 40: 「エスカレート」のシミュレーション結果

の成果」は 40、「Ｂの成果」は 10 である。このギャップを埋める行動として、Ａは、「Ｂの成果」に対する「Ａの成果」を２倍に保持しようとし、Ｂは、「Ａの成果」に対する「Ｂの成果」を同等に保持しようとする。Time $= 0$ 付近では、「Ａの成果」は「Ｂの成果」の４倍あり、基準の２倍を超えるため、Ａは「Ａの活動」を減少させるとともに、Ｂは「Ｂの活動」を急激に増加させる調整が行われる。調整の結果、「Ａの成果」は「Ｂの成果」に対して２倍を下回る状況に至り、Ａは「Ａの活動」は増加させる行動に転じる。Ｂもまた、「Ｂの成果」が「Ａの成果」と同等になるまでには至っていないので、増加させ続ける。こうした関係がエスカレートしてゆき、「Ａの成果」、「Ｂの成果」とも増大し続ける。

11　共有地の悲劇

11.1　ことわざ

この基本型は、次のような状況で生じる。AさんもBさんも活発に活動し、それぞれ順調に利得を増やしてゆくが、両者の活動が次第に大きくなるに従って、共通に利用している資源（環境）の制約のために、やがて、個々の利得が減少し始める。こうした状況を克服するために、さらに両者は個々の活動を活発にするようになるが、それが結果的に活動の総計を高め、共通資源（環境）のさらなる制約に直面するようになる。

ビジネスにおける例としては、つぎのような小売り販売促進のための共同戦力利用のケースが考えられる。ある会社のA営業所は、本社のマーケッティング部門の戦力を利用して順調に売上げを伸ばすことになるが、これを聞いたB営業所も同じ戦略を採用し同様に売上げを伸ばすようになる。こうして両営業所は、販売促進のために、ますます本社のマーケッティング部門に依存するようになるが、両営業所の販売促進努力は、結果的に本社のマーケッティング部門の戦力の制約に直面し、やがて両営業所の販売努力にもかかわらず、業績が悪化し始める。

共有地の悲劇に該当することわざとして、以下が考えられる。

- 呉越同舟
  While the thunder lasted, two bad men were friends.
- 同じ穴の貉
  Birds of a feather flock together.
- 虻蜂取らず
- 二兎を追う者は一兎をも得ず

11.2　因果ループ

「共有地の悲劇」のシステム構造は、上の例のAさんとBさんの個人的活動による利益の追求の結果生じる悲劇を用いれば、図 41（左）のような因果ループ図で示すことが出来る。

次に SD モデル構築のために、ストックを見出そう。残念ながらこれまで利用してきた時間を停止する思考実験はここでは利用できない。なぜならば時間が停止すればすべての活動も止まり、その結果である利益も実現されないからである。そこで第 2 節の脚注 4 の方法を援用する。即ち元来はフロー概念でとらえるのが適切な、Aの利益、Bの利益、及び個々の活動利益をストックと見なすのである。即ちこれらをある時点に於ける活動の結果としての情報量としてストック扱いとするのである。図 41（右）は、このようにして得られたストックを示している。

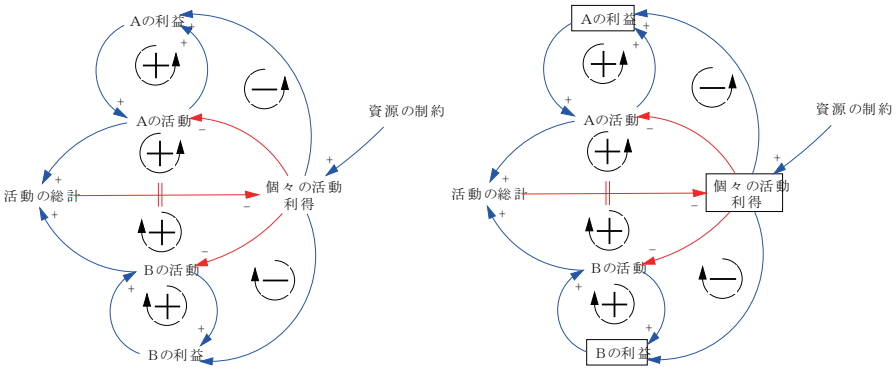

図 41: 「共有地の悲劇」の因果ループ図（左）とそのストック明 示（右）

この結果、これら３つのストックは隣接することになる。よって第２節のステップ５で論じたように、新たに変数（フロー）を挿入する必要が生じてくる。それらをＡおよびＢの利得率として追加し、図 42 で示したように、これらをＡおよびＢの利得の原因の一部と考える。

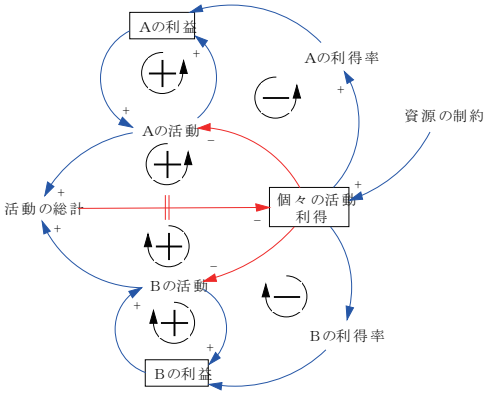

図 42: 隣接するストック図の展開

11.3 SD モデル

SD モデルを作成するために、図 42 に対して、第２節で説明した変換を加えていく。ステップ６を適用すると、図 43 のようになる。なお、ここでストック「Ａの利得」に対しては、「Ａの活動」と「Ａの利得率」の２つの矢印が正の因果関係で入っている。この２つの要因の意味を吟味すれば、別々に「Ａの利得」に対してインフローとなることは考えにくく、「Ａの活動」と「Ａの利得率」の積をとって、「Ａの利得の変化量」として、Ａの利得に対するインフローとするのが良い。「Ｂの利得」に関しても同様である。

次にこのようにして構築されたＳＤモデルのフレームワークに、必要な定数を充足して数値データを入れる。VensimPLE によるプログラミングコードを表 11 に併記する。

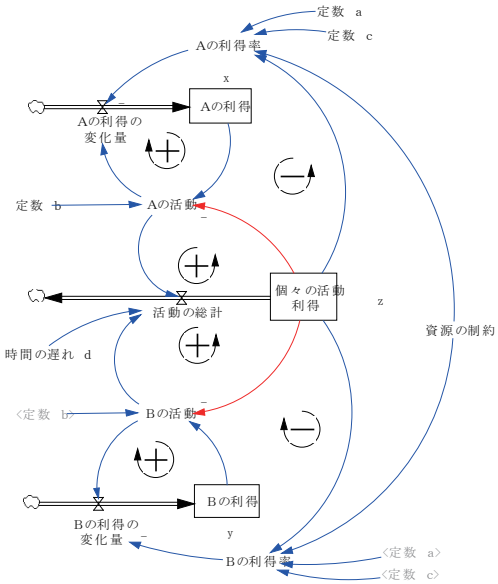

図 43: 「共有地の悲劇」の SD モデル

表 11: 共有地の悲劇　SD モデル式

個々の活動利得= INTEG (-活動の総計, 資源の制約)
定数 a = 1
定数 b = 60
定数 c = 0.6
時間の遅れ d = 5
活動の総計= DELAY FIXED (Aの活動 + Bの活動,
時間の遅れ d, Aの活動 + Bの活動)
資源の制約 = 100
Aの利得= INTEG (Aの利得の変化量, 4)
Aの利得の変化量 = Aの活動 * Aの利得率
Aの利得率= 定数 a * (個々の活動利得/資源の制約 - 定数 c)
Aの活動 = Aの利得/個々の活動利得 * 定数 b
Bの利得= INTEG (Bの利得の変化量, 3)
Bの利得の変化量 = Bの活動*Bの利得率
Bの利得率 = 定数 a * (個々の活動利得/資源の制約 - 定数 c)
Bの活動 = Bの利得/個々の活動利得 * 定数 b

11.4　微分方程式系

図 43 のＳＤモデルにおいて、Aの利得を x、Bの利得を y、個々の活動利得を z とおくと次式が得られる。なお、ここで z_0 は個々の活動利得の初期値

となる資源の制約を表す。

$$\dot{x} = ab\left(\frac{1}{z_0} - \frac{c}{z(t))}\right)x(t) \tag{16}$$

$$\dot{y} = ab\left(\frac{1}{z_0} - \frac{c}{z(t))}\right)y(t) \tag{17}$$

$$\dot{z} = -(x(t-d) + y(t-d))\frac{b}{z(t))} \tag{18}$$

初期条件: $x(0) = x_0, y(0) = y_0, z(0) = z_0$.

11.5 シミュレーション

図 44 にＳＤモデルのシミュレーション結果を示す。初期条件及び定数値は次のとおりである。$x_0 = 4, y_0 = 3, z_0 = 100, a = 1, b = 60, c = 0.6, d = 5$.

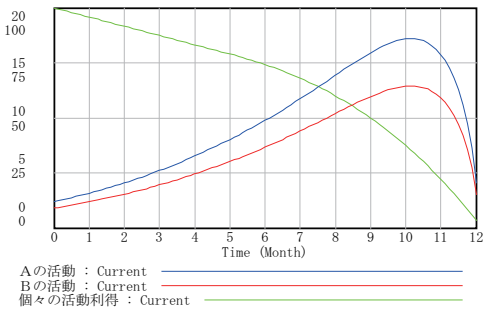

図 44: 「共有地の悲劇」のシミュレーション結果（活動）

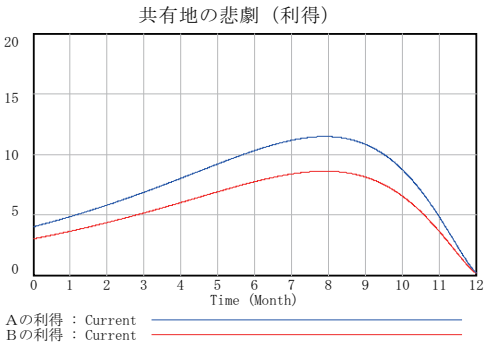

図 45: 「共有地の悲劇」のシミュレーション結果（利得）

Time = 0 から Time = 10 付近までは、A、Bとも「利得」を高めるために「活動」を増大させる。一方で、「活動の総計」の増大により、時間遅れは

あるものの、共有資源を消耗させ、「個々の活動利得」を減少させる（図 44)。「個々の活動利得」が減少するにつれ、A、Bの「利得率」もまた減少していく。Time = 0 から Time = 8 付近まではA，Bの「利得率」が正であり、その間は、A，Bとも「活動」を増加させ続けるが、Time = 8 付近以降は、A，Bとも「利得率」が負になり、A，Bともに「活動」による「利得」のフローは負に転じ、「利得」のストックもまた減少に転じる。最終的には、「利得」のストックの減少により、「活動」もまたゼロに収束する（図 45)。

12　結論

本論文では、まず因果ループからＳＤモデルを構築するための一般的方法を論じ、そこで得た導出プロセスをシステム思考の８基本型の具体的事例に応用して考察してきた。そこから明らかになったことを以下にまとめる。（１）ＳＤモデルの構築においては、因果ループの要素に示された変数のうちから、ストックを識別することが重要である。ストックを見分ける方法としては、時間を停止する思考実験が有効である。（２） 因果ループ図において、ストックの特定ならびに、その意味付けが確立した後は、半ば半自動的に因果ループ図をＳＤモデル図に変換できる。（３）同様にこうして構築された SD モデルを、非線形微分方程式体系に変換するのも、ＳＤモデリングに対する多少の熟練を必要とするものの、半ば半自動的作業となる。

システム基本型のような因果ループ図をＳＤモデルに変換するプロセスは、対象となるシステムをモデル開発者が如何に認識するかという重要なプロセスである。特に、ストックを如何に認識するかは、ＳＤモデルを利用したソリューションの導出の成果にも直結するものであり、重要である。また、時間の遅れに関する考察も、その事象の発生メカニズムに対する深い考察なしにＳＤモデルを導出することは出来ない。因果ループ図からＳＤモデルを導出するプロセスは、対象となるシステムに対する深い洞察に基づいてなされるべき重要なプロセスである。

そこでこうしたシステムの理解の一助となるものとして、先人が蓄積した知恵が凝縮されたことわざが有用な役割を果たすのではとの思いで、システム思考の基本型とことわざを関連づける試みにも挑戦した。人類の英知が因果ループ図として定性的に表現され、それがさらにシミュレーション可能な SD モデルとして、定量的に人類の未来を切り開く。本論文はそうした「システム知の創造」への一里塚となることを願っている。

参考文献

[1] Rafael E. Bourguet-Diaz, Gloria Perez-Salazar, 2003. On Mathematical Structures for Systems Archetypes, Proc. 21 International Conference

of the System Dynamics Society, P.I.Davidsen, E.Mollona, V.G.Diker, R.S.Langer, J.I.Rowe, ed., SDS, pp.1-11.

[2] Daniel H. Kim and Virginia Anderson, 1998. Systems Archetype Basics - From Story to Structure, Pegasus Communications, Inc. MA, USA.
システム・シンキング トレーニングブック、宮川雅明・川瀬誠 訳、 日本能率協会マネジメントセンター、２００２年

[3] George P. Richardson, 1986. Problems with Causal-Loop Diagrams, System Dynamics Review, Volume 2, Number 2, Summer 1986.

[4] Barry Richmond, 2000. The “ Thinking ” in Systems Thinking: Seven Essential Skills, Pegasus Communication, Inc., First edition, ISBN 1-883823-48-X

[5] Peter M. Senge, 1990. The Fifth Discipline - The Art and Practice of The Learning Organization - Currency Doubleday, New York.
最強組織の法則、新時代のチームワークとは何か、守部 信之（訳）徳間書店、１９９５年

[6] Peter M. Senge, Art Kleiner, Charlotte Roberts, Richard B. Ross, Bryan J. Smith, 1994. The Fifth Discipline Fieldbook - Stragegies and Tools for Building a Learning Organization, Currency Doubleday, New York.
フィールドブック　学習する組織「５つの能力」、　日本経済新聞社、２００３年

[7] John D. Sterman, 2000. Business Dynamics - Systems Thinking and Modeling for a Complex World, Irwin MaGraw-Hill, New York.

[8] Kim Warren, 2002. Competitive Strategy Dynamics, Wiley.

[9] Kaoru Yamaguchi, 1999. Stock-Flow Fundamentals, Delta Time(DT) and Feedback Loop -From Dynamics to System Dynamics-, the 17th International Conference of the System Dynamics Society: System Thinking for the Next Millennium, New Zealand, July 20-23.

[10] 西村行功, 2004.　システム・シンキング入門、　日経文庫.

[11] 山口　薫、2007. システム思考の８基本型（モジュール４）、経営システムダイナミックス講義ノート、同志社ビジネススクール.

[12] 改訂新版読み・書き・話す　故事ことわざ辞典 A Dictionary of Proverbs & Sayings、学習研究社、１９９９年