

# 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

システムダイナミクス (System Dynamics, SD) はMITのジェイ・フォレストラー (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、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創始者のフォレストラー名誉教授が、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関係、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コア (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て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に基づいてマーケットの趨勢を理解し、不確実な未来へ備える戦略立案に応用している。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<sup>1</sup>。

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。

---

<sup>1</sup> 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 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（矢頭を+の記号で表示）。

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判断し、赤色の矢印で結ぶ（矢頭を-の記号で表示）。図1は、15のキーワードから6つの因果関係を見いだした図を表している<sup>2</sup>。

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、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、システムを指数的に (Exponentially) 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

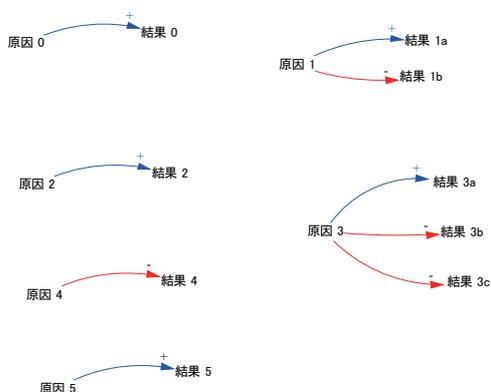


図1: キーワードの因果関係

<sup>2</sup>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モデル作成ソフトは、これ以外にも STELLA, iThing, 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

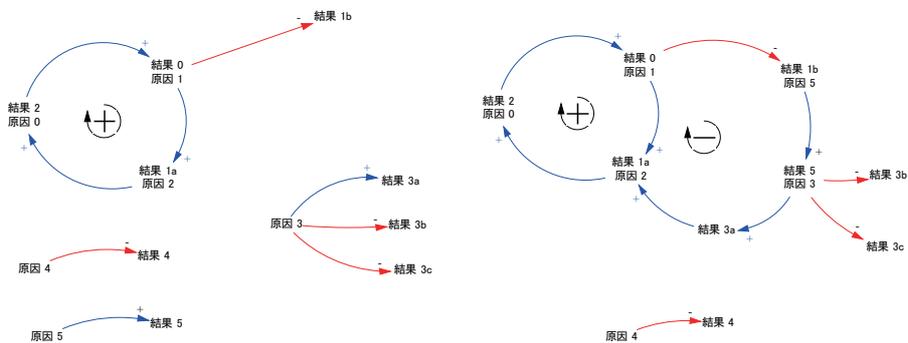


図 2: 因果ループ (左) のその拡張 (右)

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が完結する (Close 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。

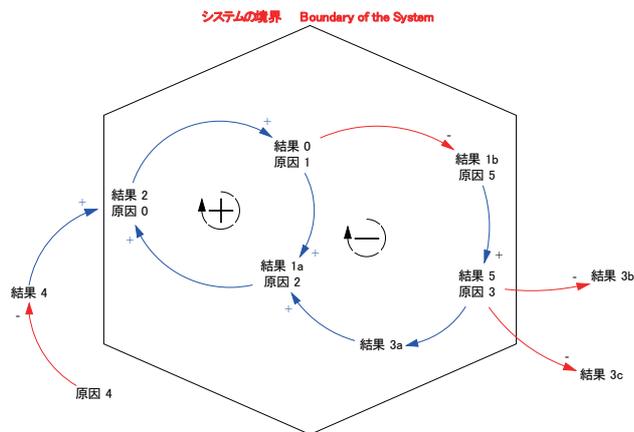


図 3: 因果ループ図

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、フィード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てき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った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

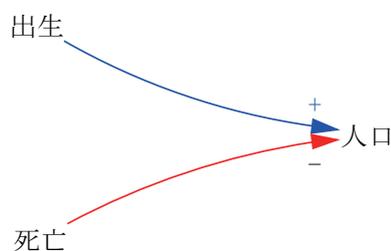


図4: 因果関係

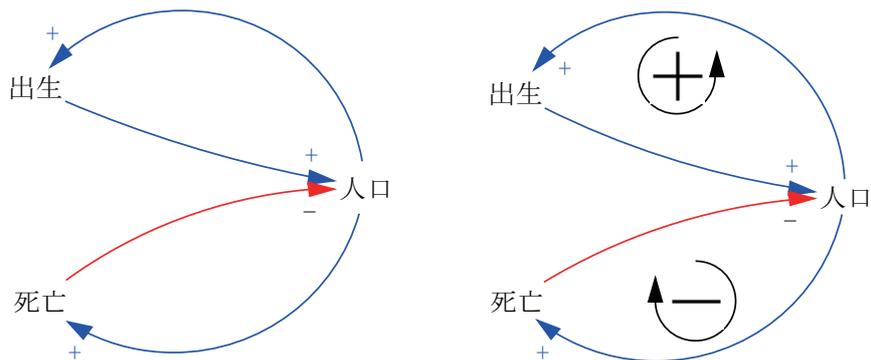


図5: 因果ループ(左)と正・負のフィードバックループ(右)]

このようにして、最も簡単な人口動態システムは、おのおのの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は判明した。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



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<sup>5</sup>。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。

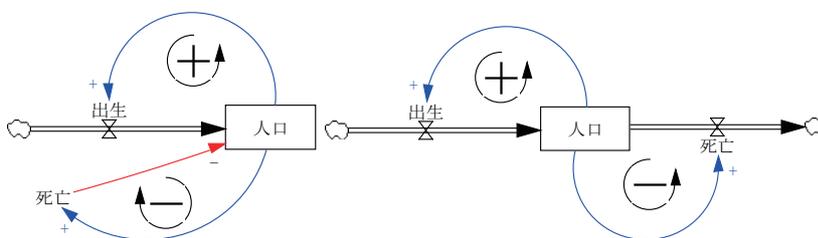


図7: インフロー（左）とアウトフロー（右）に置換

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。

<sup>5</sup>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](http://www.muratopia.org/JFRC)) に掲載の入門マニュアル等を参照されたい。

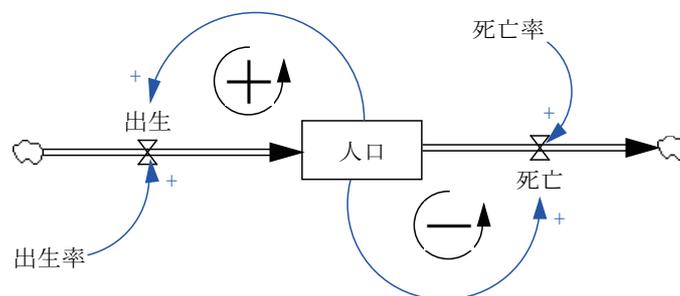


図 8: 数値データ SD モデルへの拡張

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。

## 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。

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

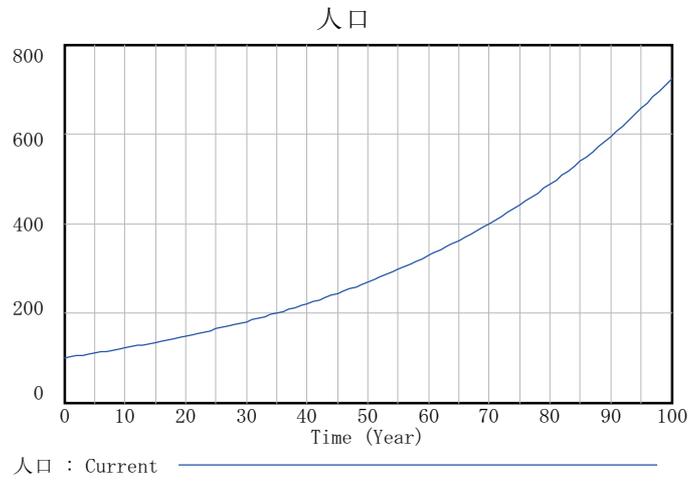


図 9: 人口モデル」のシミュレーション結果

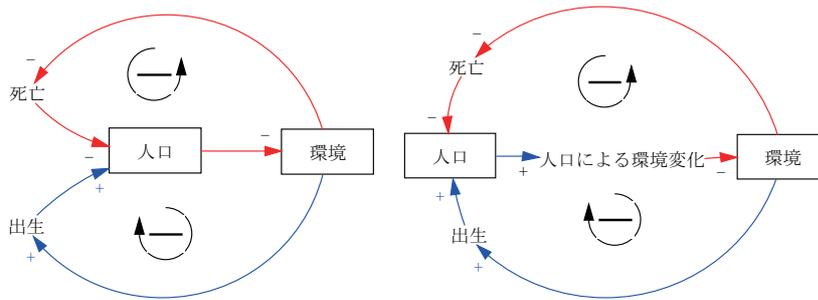


図 10: 隣接するストック同士（左）の間に要素を追加（右）

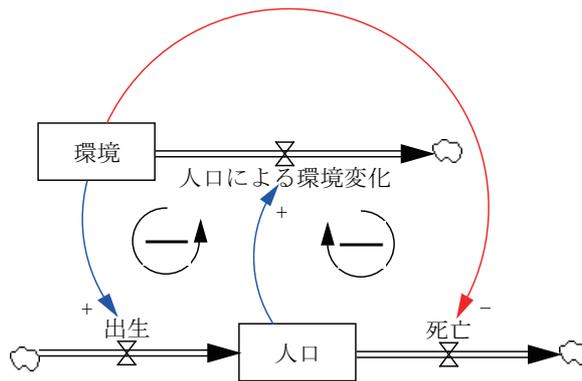


図 11: SDモデルの構築

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」について上述のス

ステップ5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つ以上見だし、ストックとして外部を四角形で囲む。(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(図7(左))し、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(図7(右))する。

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<sup>6</sup>。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。

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ダイナミックな振る舞いの中で、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

<sup>6</sup> 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。

型のものとしてとらえられるというシステム的な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によって、複雑化した問題が単純化して捉えられるようになり、様々な振る舞いをもたらすシステムの構造が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、還元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<sup>7</sup>。「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。

---

<sup>7</sup>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、サンフランシスコの金門橋を渡った山間部にあるグリー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、筆者の一人にある国際SD学会で語ってくれた。この禅センターは筆者の一人が留学時代に頻り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。

## 4 応急処置の失敗

### 4.1 応急処置の失敗とことわざ

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）構造を関連づけるのは、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として経営の指針とし、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分析することになる。ここでは、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・パターンを8基本型として考察し、それらに該当することわざを探り出してみよう<sup>8</sup>。

まずは、以下のことわざを観察してみよう<sup>8</sup>。

- 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ルを張り替えて再出荷、客が残した食材を使い回すと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。

---

<sup>8</sup>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] 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、彼らの英知に感謝！。

## 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」のダイナミクス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、「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。

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。

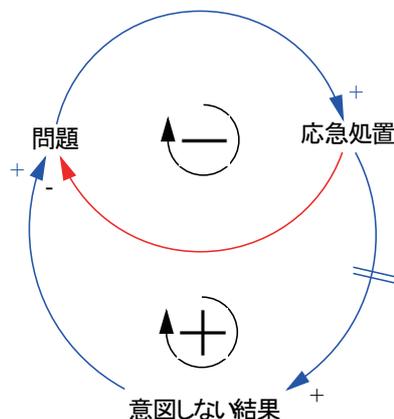


図 12: 「応急処置の失敗」の因果ループ図

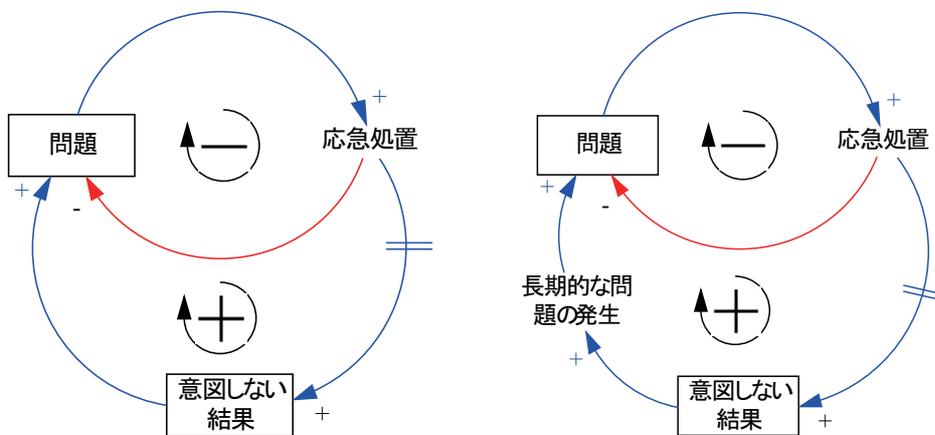


図 13: ストック明示の因果ループ図

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、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。

### 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、第 2 節で説明した変換を加えていく。第 2 節のステップ 5 を、ストック「問題」と「意図しない結果」に適用すると、両者ともに正の因果関係となっているので、図 14 のような SD モデルが構築できる。

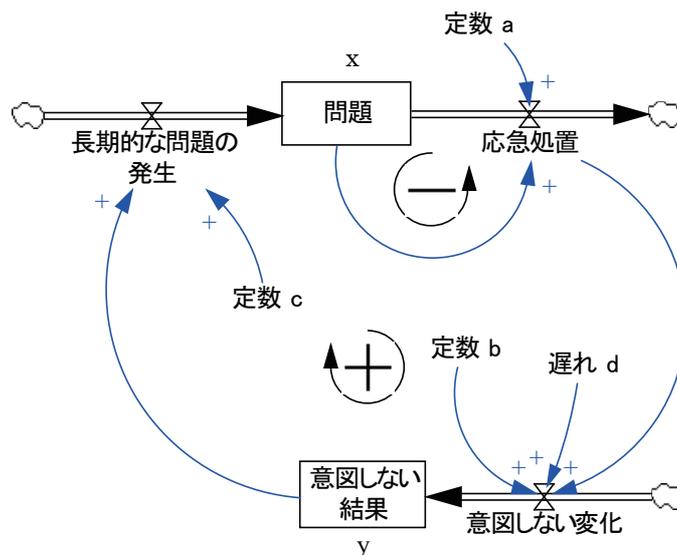


図 14: 「応急処置の失敗」の SD モデル

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インフロー（=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<sup>9</sup>。

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

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

初期条件  $x(0) = x_0, y(0) = y_0$

## 4.5 シミュレーション結果

図15にシミュレーション結果を示す。条件は次のとおり。 $x_0 = 100, y_0 =$

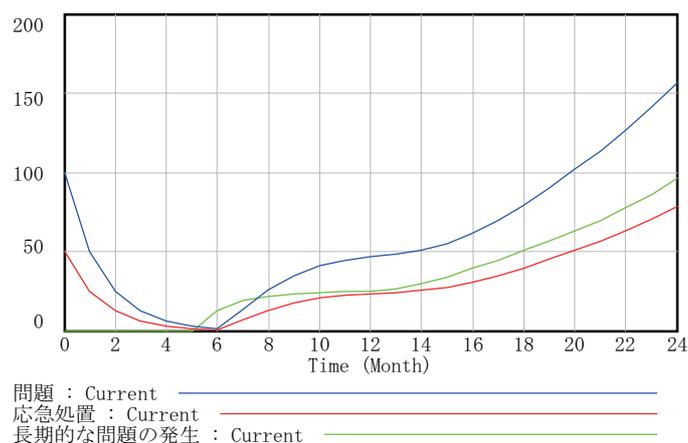


図15: 「応急処置の失敗」のシミュレーション結果

0,  $a = 0.5, b = 0.5, d = 5$

## 5 問題の転嫁

### 5.1 問題の転嫁とことわざ

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

<sup>9</sup>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。

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」を解決するための2種類のメカニズムが存在する。一つ目は短期的視点での打ち手（即ち「応急処置」）をすることにより、「問題」を軽減させるというものである。二つ目は、長期的視点での打ち手（即ち「根本的解決」）をすることにより、時間の遅れを伴って「問題」を軽減させるというものである。しかしながら、「応急処置」は「意図しない結果」を生み、「根本的解決」を阻害する。「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、これに反して「問題の転嫁」では、問題の根本的解決を図るサブシステムが新たに加わることになる。ところが、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」の因果ループを図16（左）に示す。

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。

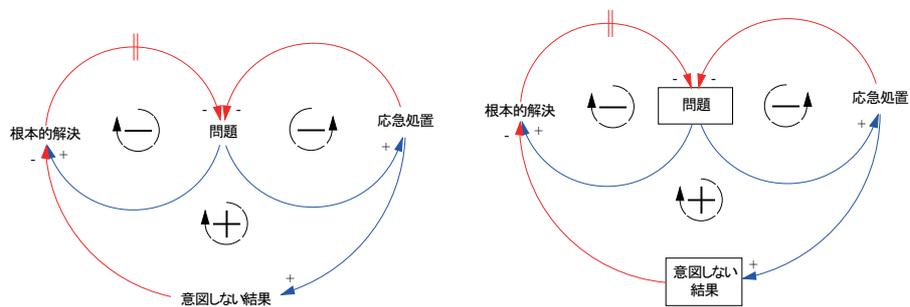


図 16: 「問題の転嫁」の因果ループ図 (左) とそのストック明示 (右)

### 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5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。

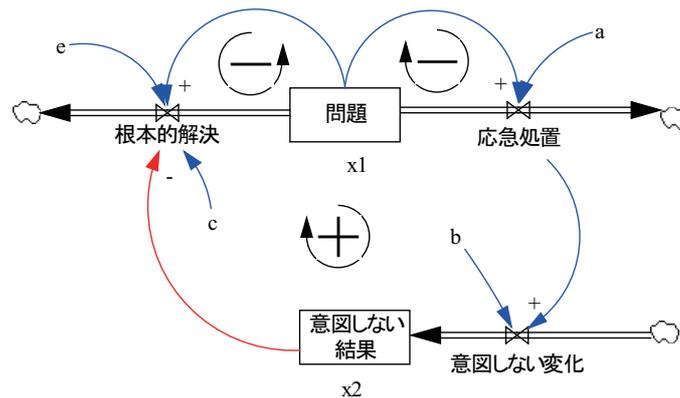


図 17: 「問題の転嫁」のSDモデル

VensimPLE によるプログラミングコードを表3に併記する。

### 5.4 微分方程式系

図17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。

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

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

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

初期条件  $x(0) = x_0, y(0) = y_0$

### 5.5 シミュレーション結果

図 18 にシミュレーション結果を示す。

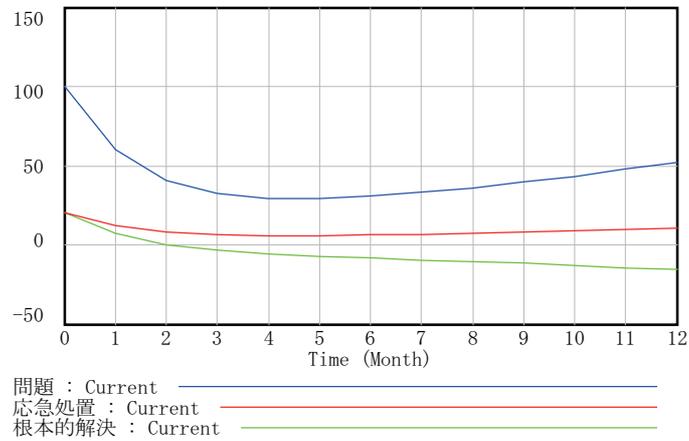


図 18: 「問題の転嫁」のシミュレーション結果

条件は次のとおり。  $x_10 = 100, x_20 = 0, a = 0.2, b = 0.5, c = 0.5, e = 0$

## 6 成長の限界

### 6.1 成長の限界とことわざ

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) 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、企業は成長・拡大することになるが、その拡大過程で経営能力の制約のために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。

事業拡大が行き詰まる例として、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」の指数関数的増加をもたらす「生成」メカニズムと、「成果」の指数関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関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、人口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・アウトフロー均衡の場合）、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)（イン・アウトフロー均衡の場合）

### 6.2.1 因果ループ

「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。こうした課程で一方、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。

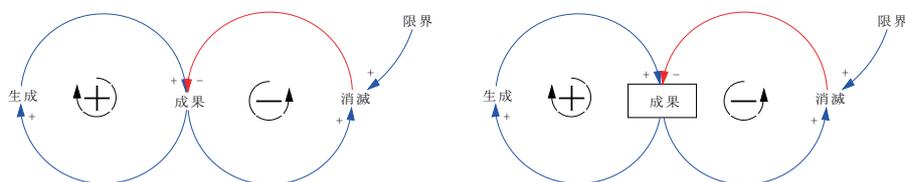


図 19: 「成長の限界(1)」の因果ループ図(左)とそのストック明示(右)

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消滅するからである。ストックを図示した因果ループ図を図19(右)に示す。

### 6.2.2 SDモデル

SDモデルを構築するために、図19(左)に対して、第2節で説明した変換を加えていく。ステップ5を、ストック「成果」に適用すると、図20の結果となる。

VensimPLEによるプログラミングコードを表4に併記する。

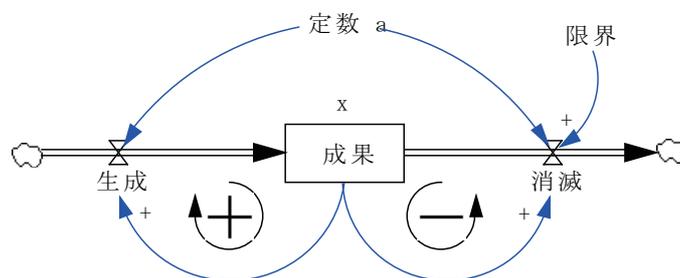


図 20: 「成長の限界 (1)」の SD モデル

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、ストック「成果」を  $x$ 、限界を  $L$  とおくと、次式が得られる。

$$\begin{aligned} \dot{x} &= ax(t) - ax(t) \left( 1 - \frac{L - x(t)}{L} \right) \\ &= a \left( 1 - \frac{x(t)}{L} \right) x(t) \end{aligned} \quad (5)$$

初期条件:  $x(0) = x_0$

### 6.2.4 シミュレーション結果

図 21 に SD モデルのシミュレーション結果を示す。条件は次のとおり。  
 $L = 100, a = 0.1$

## 6.3 成長の限界 (2) (インフローがゼロに収束する場合)

### 6.3.1 因果ループ

「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、成果と成果余地の積に比例して成果

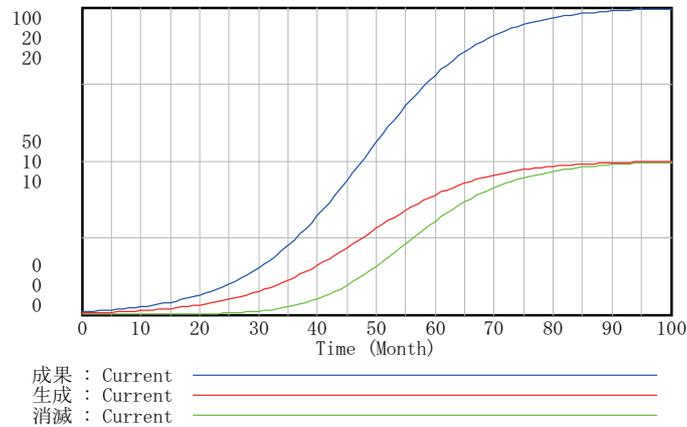


図 21: 「成長の限界 (1)」のシミュレーション結果

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果の量のフィードバックが優勢であり、成果の量は指数関数的に増加する。しかし、成果の量が限界に近づく末期では、成長余地の減少が優勢となり、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。

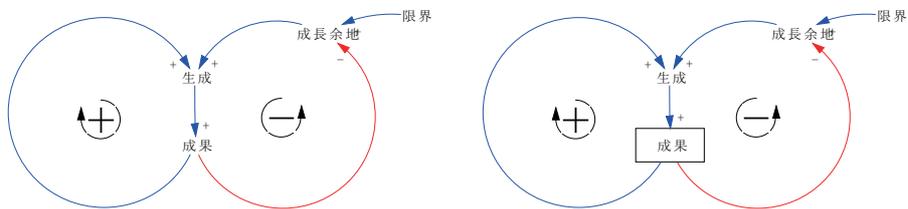


図 22: 「成長の限界 (2)」の因果ループ図 (左) とそのストック 明示 (右)

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 5 を、ストック「成果」に適用すると、図 23 の結果となる。

VensimPLE による プログラミングコードを表 5 に併記する。

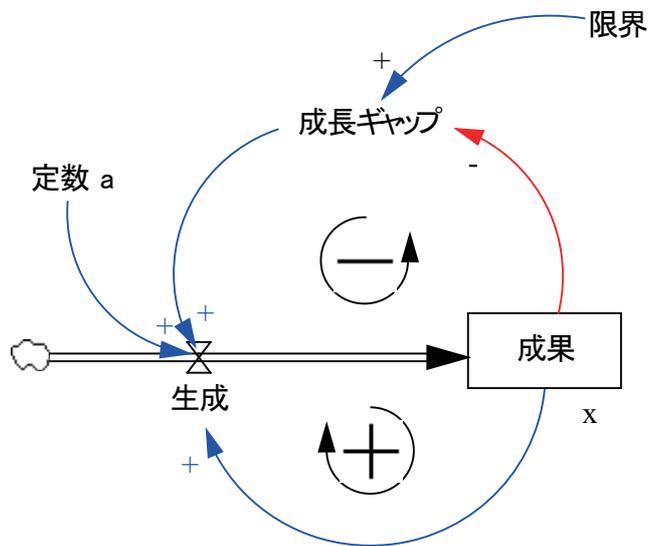


図 23: 「成長の限界 (2)」の SD モデル

表 5: 成長の限界 (2) SD モデル式

定数 $a = 0.1$ 成果 = INTEG (生成, 1) 成長余地 = (限界 - 成果) / 限界 生成 = 定数 $a$ * 成長余地 * 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、ストック「成果」を  $x$ 、限界を  $L$  とおくと、次式が得られる。

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

初期条件:  $x(0) = x_0$

### 6.3.4 シミュレーション結果

図 23 の SD モデルのシミュレーション結果を示す。条件は次のとおり。  
 $L = 100, a = 0.1, x_0 = 1$ .

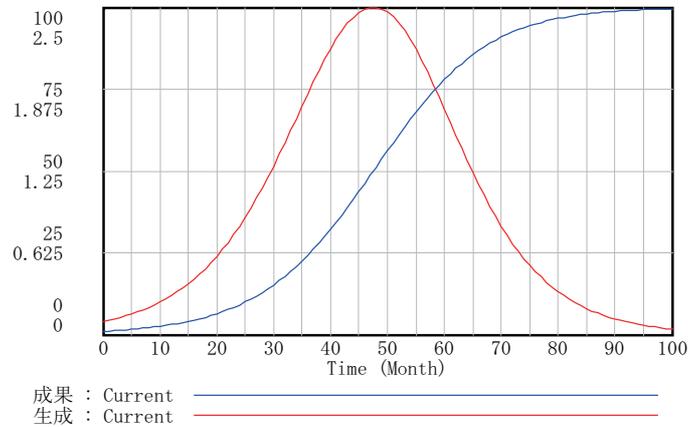


図 24: 「成長の限界（2）」のシミュレーション結果

## 6.4 成長の限界（3） 増減ストック総和が一定の場合

### 6.4.1 因果ループ

「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（成果余地）のような関係が該当する。成長の限界は、成果と成果余地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、成果余地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、成果と成果余地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。

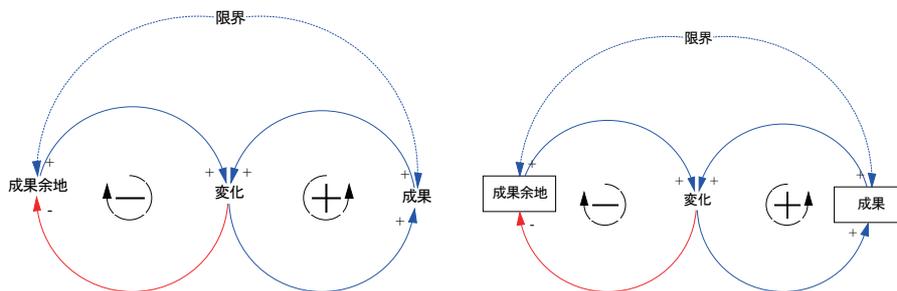


図 25: 「成長の限界（3）」の因果ループ図（左）とそのストック明示（右）

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「成果余地」となる。なぜならば、時

間を止めても、成果と成果余地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 5 を、ストック「成果」に適用すると、図 26 の結果となる。

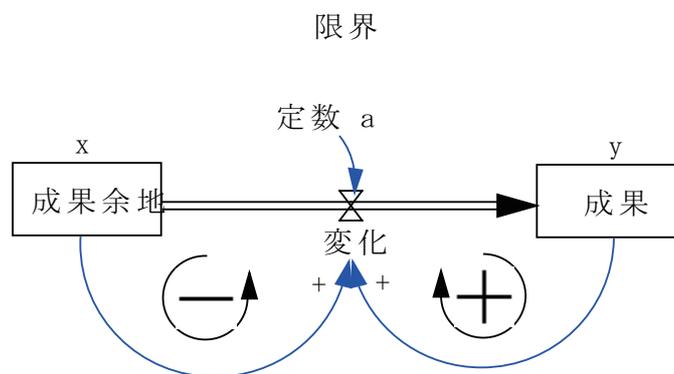


図 26: 「成長の限界 (3)」の SD モデル

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 * 成果余地
成果 = INTEG (変化, 1)
成果余地 = 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、ストック「成果」を  $y$ 、ストック「成果余地」を  $x$  とおき、限界を  $L = x + y$  と定義すると次式が得られる。

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

初期条件:  $x(0) = x_0$

#### 6.4.4 シミュレーション結果

図 27 に SD モデルのシミュレーション結果を示す。条件は次のとおり。  
 $L = 100, a = 0.001, x_0 = 1$

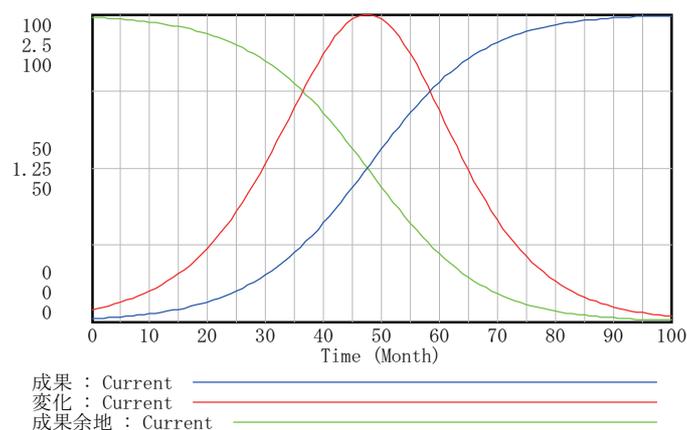


図 27: 「成長の限界 (3)」のシミュレーション結果

## 7 目標のなし崩し

### 7.1 目標のなし崩しとことわざ

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 turning 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。目標と成果の差であるギャップがダイナミクスの源である。ギャップが生じた場合それを埋めるために2つのメカニズムが存在する。一つ目は改善行動により成果を大きくすることによりギャップを埋めるメカニズムである。二つ目は目標を修正してギャップを埋めるメカニズムである。「目標のなし崩し」の因果ループ図を図28(左)に示す。

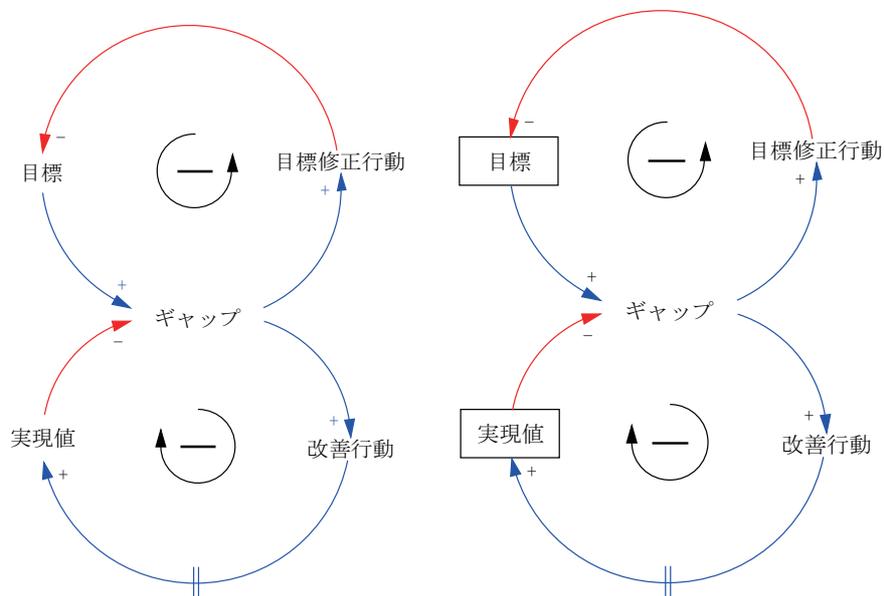


図 28: 「目標のなし崩し」の因果ループ図(左)とそのストック明示(右)

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、上役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、上役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と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、設定を変更するという意味で、一種の組織能力の反映としての目標であると考えて。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す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 5 を、ストック「成果」に適用すると、図 29 の結果となる。

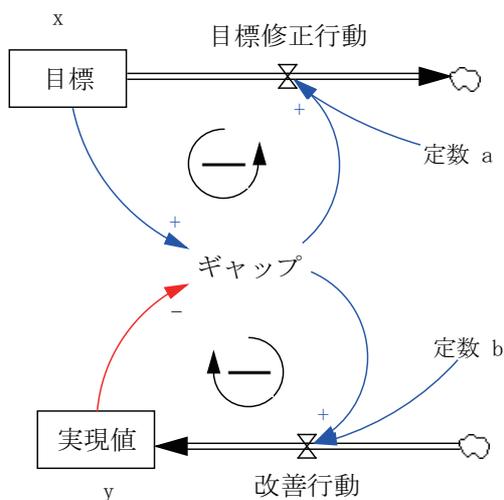


図 29: 「目標のなし崩し」の SD モデル

VensimPLE によるプログラミングコードを表 7 に併記する。

表 7: 目標のなし崩し SD モデル式

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

## 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。

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

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

初期条件:  $x(0) = x_0, y(0) = y_0$

## 7.5 シミュレーション結果

図 30 に SD モデルのシミュレーション結果を示す。条件は次のとおり。  
 $x_0 = 100, y_0 = 50, a = 12, b = 24$

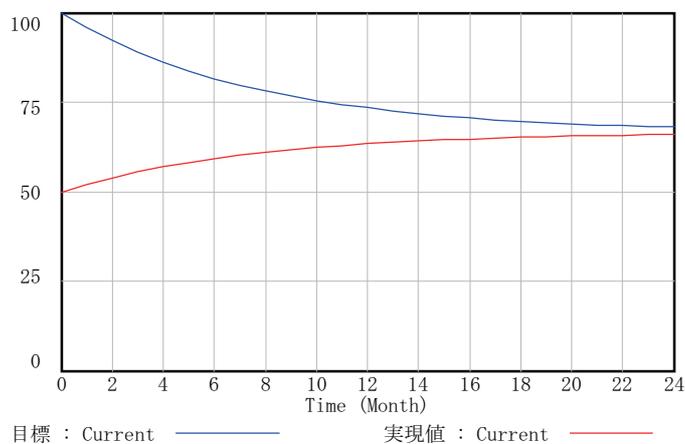


図 30: 「目標のなし崩し」のシミュレーション結果

## 8 成長と投資不足

### 8.1 成長と投資不足とことわざ

「成長と投資不足」の基本型は「成長の限界」の構造を一部含んでいる。すなわち、パフォーマンスの制約によって成長の限界に直面しはじめるという点であるが、その限界を適切な投資によって引き上げ、成長を持続させるこ

とが出来るといえるというのが新たに加わる。ところが往々にしてこの投資を怠り、結果的に投資不足となって成長が止まってしまうというのである。

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（オーバークッキング、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）縄を縛う

## 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」に対して生成メカニズムが存在する。また、成果が生産能力を上回れば品質問題が増加し、成果を抑制するメカニズムが存在する。これは成長の限界と類似のメカニズムである。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限界をアップできる点が成長の限界とは異なる。しかしながら、一般的に能力開発への投資は長期に及び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「能力」となる。成果はその背後に生成のメカニズムを含んでいる。また、能力は投資による能力開発のメカニズムを含んでいる。成果と能力の差が品質となる。品質は、成果に対しても能力に対してもネガティブフィードバックされる。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 5 を、ストック「成果」に適用すると、図 32 の結

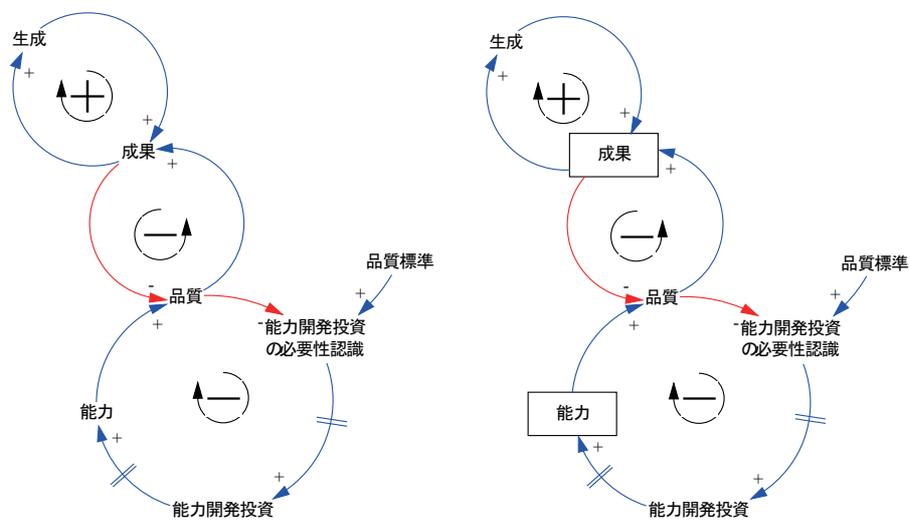


図 31: 「成長と投資不足」の因果ループ図 (左) とそのストック 明示 (右)

果となる。

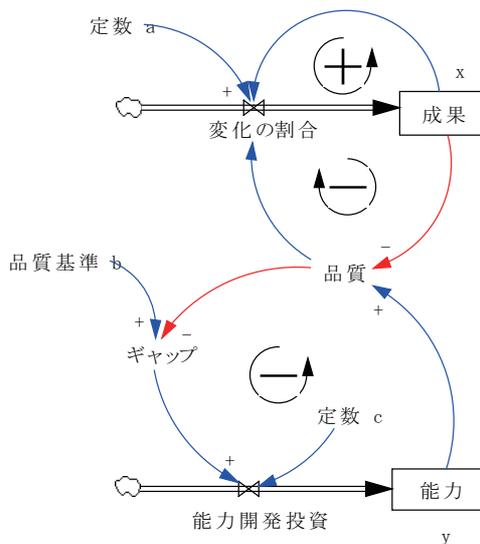


図 32: 「成長と投資不足」の SD モデル

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。

表 8: 成長と投資不足 SD モデル式

定数 a = 0.01
定数 c = 60
ギャップ = 品質基準 - 品質
品質 = 能力 - 成果
品質基準 b = 20
変化の割合 = 定数 a * 成果 * 品質
成果 = INTEG (変化の割合, 1)
能力 = INTEG (能力開発投資, 100)
能力開発投資 = ギャップ / 定数 c

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

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

初期条件:  $x(0) = x_0, y(0) = y_0$

### 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。条件は次のとおり。  
 $x_10 = 1, x_20 = 100, a = 0.01, b = 20, c = 60$

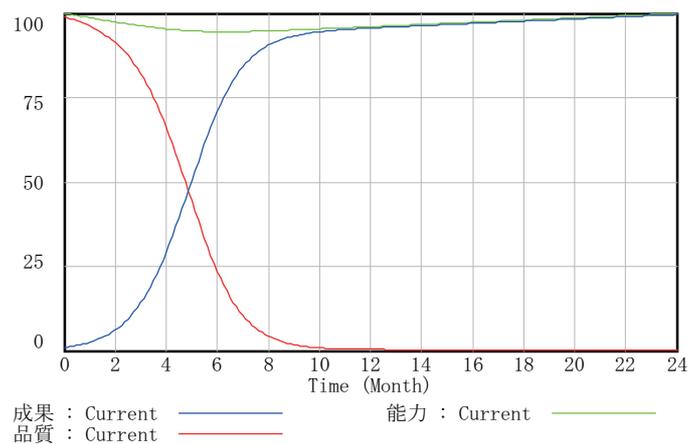


図 33: 「成長と投資不足」のシミュレーション結果

## 9 成功には成功

### 9.1 成功には成功とことわざ

「成功には成功」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始める。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、やがら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。こうした事例はいわゆる勝ち組・負け組の基本的な構造となる。

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リソースを2つの活動が取り合うシステムである。リソースの配分は、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。

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した信用である。一方でフローは、リソースの配分の付け替えである。リソースは総量が一定であるので、一方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。

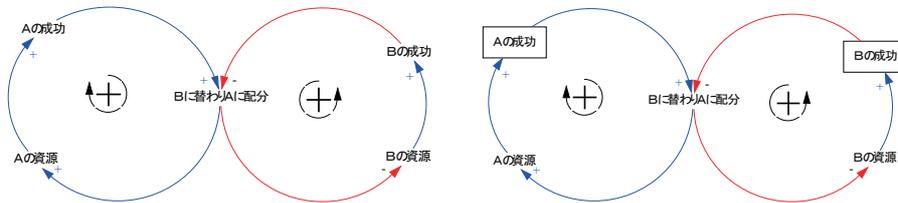


図 34: 「成功には成功」の因果ループ図 (左) とそのストック明示 (右)

### 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 5 を、ストック「成果」に適用すると、図 35 の結果となる。

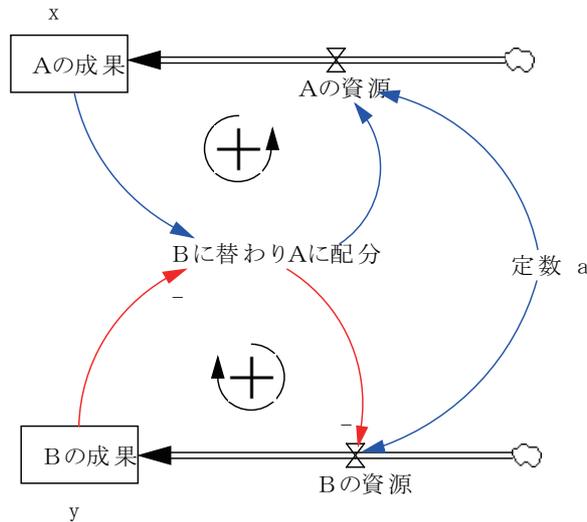


図 35: 「成功には成功」の SD モデル

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。

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

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

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
---

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。条件は次のとおり。  
 $x_0 = 5.5, y_0 = 4.5, a = 0.1$

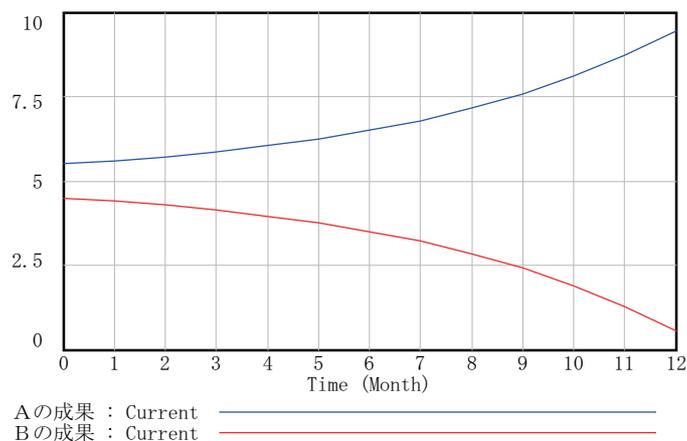


図 36: 「成功には成功」のシミュレーション結果

## 10 エスカレート

### 10.1 エスカレートとことわざ

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、 $a$  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ったことが、他方に脅威となり防御的に成果を増加させ、それがまた他方の脅威となるというダイナミクスをたどる。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の増加の活動が繰り返される。なお、「成功には成功」はリソースが一定であったが、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）に示す。この因果ループ図では、2つの均衡ループが描かれるが、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、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。

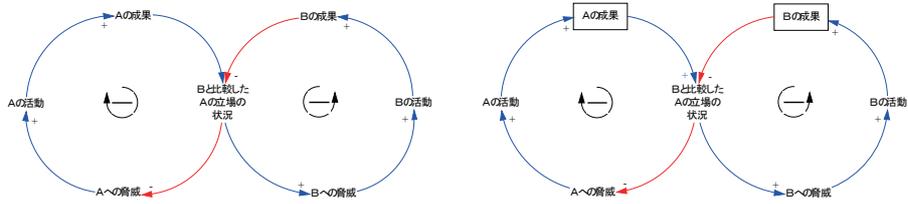


図 37: 「エスカレート」の因果ループ図(左)とそのストック明示(右)

このエスカレートの8の字型の因果ループ図を、Untwisted Structureの構造で書き換えると、ひとつの正のフィードバックループとなる。競争的状况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と考えられるからである。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。

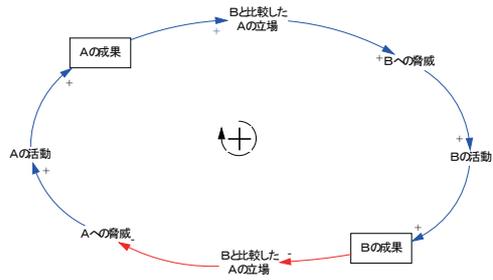


図 38: 8字型因果ループの展開図

### 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5を、ストック「成果」に適用すると、図39の結果となる。

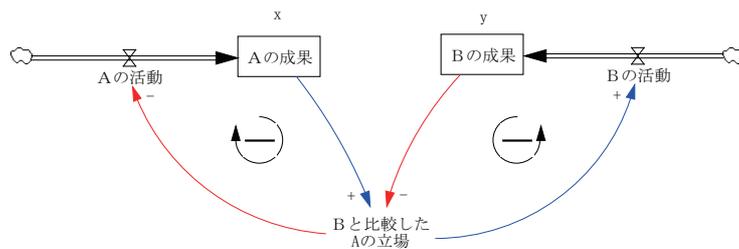


図 39: 「エスカレート」のSDモデル

VensimPLEによるプログラミングコードを表10に併記する。

### 10.4 微分方程式系

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

表 10: エスカレート SD モデル式

Aの成果 = INTEG (Aの活動, 40)
Aの活動 = 2-Bと比較したの立場
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 = -(1-Bと比較したの立場)

$$\dot{x} = a \left( R_a - \frac{x(t)}{y(t)} \right) \quad (14)$$

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

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。条件は次のとおり。

$x_0 = 40, y_0 = 10, R_a = 2/1, R_b = 1/1, a = 1, b = 1$

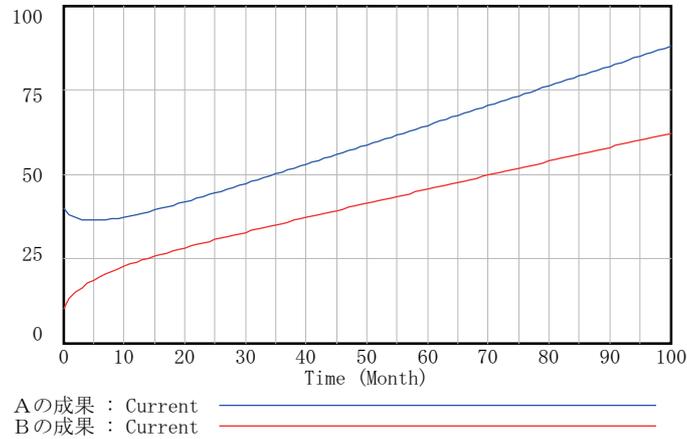


図 40: 「エスカレート」のシミュレーション結果

## 11 共有地の悲劇

### 11.1 共有地の悲劇とことわざ

「共有地の悲劇」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、さらに両者は個々に活発に活動しあうことになるが、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ことになる。

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。即ちこれらがある時点に於ける活動の結果として

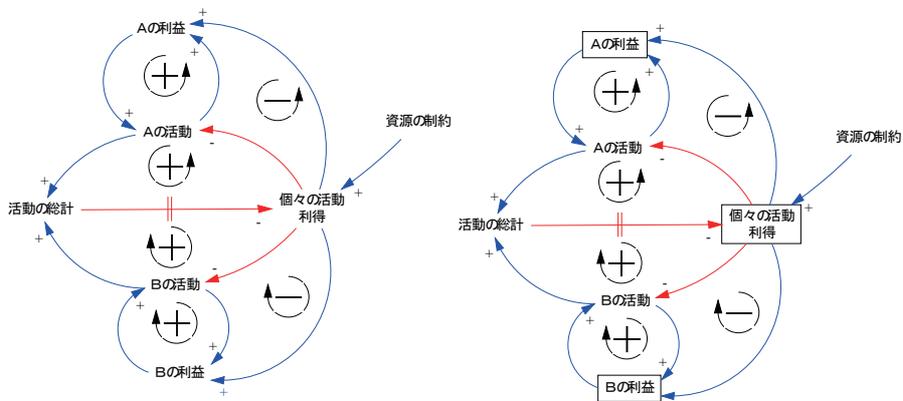


図 41: 「共有地の悲劇」の因果ループ図（左）とそのストック明示（右）

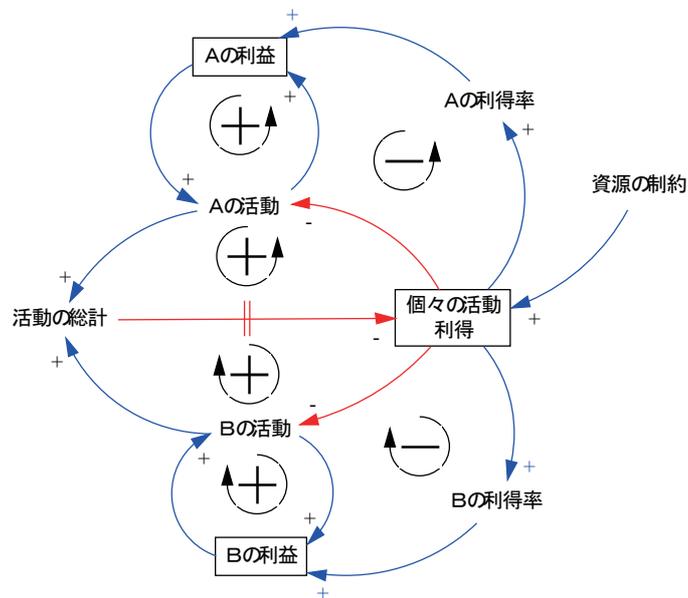


図 42: 隣接するストック図の展開

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。

この結果、これら 3 つのストックは互いに隣接することになる。よって第 2 節で論じたように、新たに変数（フロー）を挿入する必要性が生じてくる。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 の原因と考える。

### 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 5 を適用すると、図 43 のようになる。

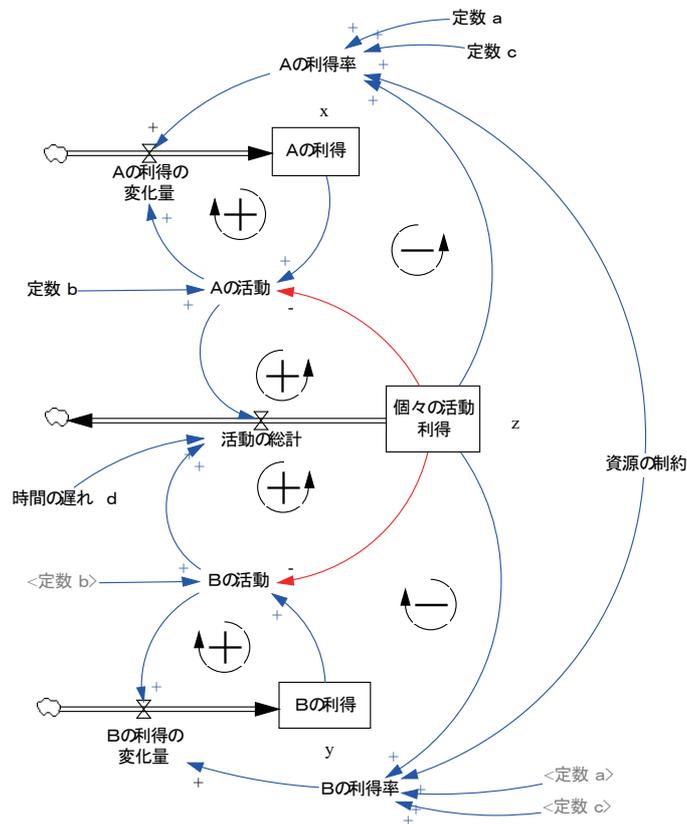


図 43: 「共有地の悲劇」の SD モデル

VensimPLE によるプログラミングコードを表 11 に併記する。

### 11.4 微分方程式系

図 30 の SD モデルにおいて、A の累積利益を  $x$ 、B の累積利益を  $y$ 、活動あたりの利得を  $z$  とおくと次式が得られる。なお、ここでは  $z_0$  資源の制約である。

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

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

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 = 0.4
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の活動の変化量, 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の活動の変化量, 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

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

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。条件は次のとおり。  
 $x_0 = 20, y_0 = 15, z_0 = 100, c = 0.4, d = 5$ .

## 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

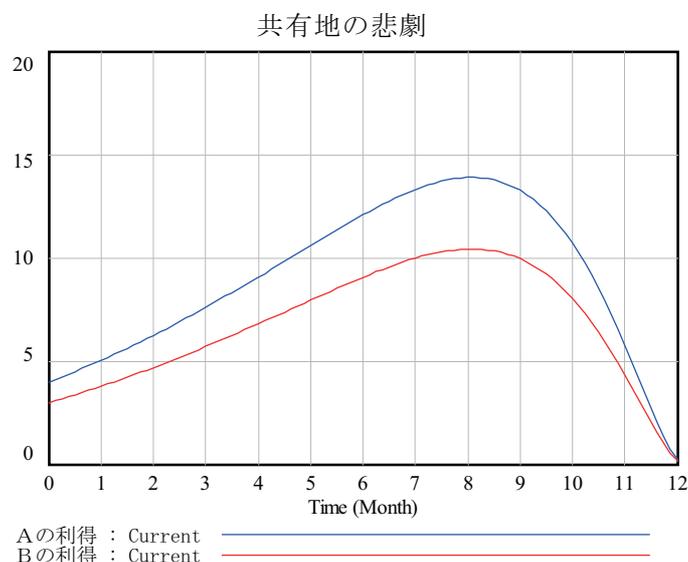


図 44: 「共有地の悲劇」のシミュレーション結果

ブをSDモデル図に変換できる。(3)同様にして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。本論文はそうした新しい「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 - Strategies 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, 2003. シナリオ・シンキング, ダイヤモンド社.
- [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クス講義ノート、同志社ビジネススクール.