論　　文

システム・ダイナミクス

～複雑系を分析するための統合的アプローチ～

高橋 裕 *

＜要旨＞

本稿では、コンピュータシミュレーションを用いた分析手法であるシステム・ダイナミクスについて説明する。システム・ダイナミクスは、戦略と政策設計のためのコンピュータ支援アプローチであり、複雑で動的なシステムに直面した際の意思決定を支援することを目的としている。はじめにシステム・ダイナミクスの概要と特徴について説明する。その後、モデル構築から分析・施策提案に至るプロセスについて詳細に説明する。特に、近年の研究コミュニティにおける議論から得られたシステム・ダイナミクスならびにコンピュータシミュレーションの利用に関する注意点について説明する。最後に施策の提案はステークホルダーに理解され実行されてこそ意味があるため、その段階における要諦についても説明する。単なる不確実性ではなく、複雑なシステムの構造に起因する様々な社会の課題に対応するため、システム・ダイナミクスはさらに利用されるべき分析手法と言える。

JEL Classification Codes：C60, C63, C69

Keywords： 複雑系、コンピュータ・シミュレーション、システム思考

* 高橋 裕：専修大学商学部教授。

System Dynamics:
An Integrated Approach to Analyzing Complex Systems

By Yutaka TAKAHASHI[§]

Abstract

This paper presents a background and an analytical approach to understanding system dynamics. System dynamics provides a computer-assisted strategy and policy design approach, aimed at supporting decision-making in the face of complex and dynamic systems. The paper starts by outlining the framework and key characteristics of system dynamics. It then details the process from model construction to analysis and policy proposals. In particular, it highlights critical consideration for utilizing system dynamics and computer simulations, drawing on recent discussions in the research community. The paper concludes by arguing that system dynamics should be more broadly applied to address the diverse social issues stemming from the structure of complex systems, rather than just from uncertainty.

JEL Classification Codes: C60, C63, C69
Keywords: complex system, computer simulation, systems thinking

[§] School of Commerce, Senshu University

1．はじめに

システム・ダイナミクスとは戦略と政策設計のためのコンピュータ支援アプローチである[1]。その利用の主な目的は、複雑で動的なシステムに直面したときに、より良い意思決定の支援を行うことである。このアプローチの特徴は、システム思考（検討対象をシステムとして捉える思考）に立脚し、フィードバックループ構造（因果関係の循環構造）に着眼してモデリングと分析を行う点にある。適用対象は広範囲に及び、複雑な社会システム、経営システム、経済システム、生態系システムを含む多くの領域で発生する、あるいは潜在的に存在すると考えられる動的な課題の分析に利用できる。

システム思考をベースにした方法論・アプローチは多数ある。その多くは定性的分析や概念整理を行うものである。これに対してシステム・ダイナミクスは、概念化のプロセスも包含しつつ、主要な分析に定量的モデルとそれを用いたシミュレーションを不可欠としている点が特徴と言える。

Sargent et al.（2017）は Wagner（1969, p. 890）がシミュレーションについて「他のすべての分析手法が適用できない場合」「シミュレーションは最後の手段である」と記述したことに触れて、このことがシミュレーションを積極的に用いることに関して常に議論を呼んできたとしている。しかし、現代社会に残された、あるいは新たに発生して解決が待たれている問題は多様な当事者・複雑に結合されたシステムによって引き起こされているのであり、解析的に（式の変形・展開によって）最適解がえられるものではないことが多い。この状況ではむしろ、システム・ダイナミクスのような整理された方法論を持つシミュレーション手法を積極的に利用すべき時代になったとも考えられる。

なお、日本においてはシステム思考をサイバネティクスの視点で語ることが多くみられる。しかし、Richardson（1991, p. 93）の指摘するとおり、システム・ダイナミクスはサーボメカニズムの系譜にある。このことは、サイバネティクスの系譜にある他のアプローチとシステム・ダイナミクスとの違いを生み出しており、理解・利用の際に注意が必要である。

システム・ダイナミクスを提唱したのは MIT Sloan School に在籍した J. W. Forrester である。Forrester は 1950 年代にこの手法を考案し、インダストリアル・ダイナミクス（Forrester, 1961）として世に送り出した。その後、適用範囲を都市の問題（Forrester, 1969）や世界的な経済や人口の問題（Forrester, 1973）に関するシステム・ダイナミクスを利用した分析を公刊し、注目を浴びた。こうして適用範囲が広がるにつれて「適用対象をより一般化した名称にすべき」という考えがシステム・ダイナミクスを利用するコミュニティで広がり、現在ではシステム・ダイナミクスと呼ばれている。Forrester 自身に関する事柄とシステム・ダイナミクスの考案に至る沿革は、Lane and Sterman（2011）に詳しく述べられている。

シミュレーションの実行にあたってはコンピュータソフトウェアを必要とする。本稿執

[1] ここでの定義は国際学術団体 System Dynamics Society の定義をベースにしている。

筆時点における日本の状況を見ると、Ventana Systems 社の Vensim、isee systems 社の Stella、PathFwd 社が運営するオンラインサービスの PathFwd が広く利用されているようである。

日本におけるシステム・ダイナミクスの名称表記については揺らぎがある。これまで公刊された文献では「システム」と「ダイナミクス」の間の「・」の有るものと無いものが、また「ダイナミクス」と「ダイナミックス」の両方が用いられている。なお、本稿執筆時点で日本における学術団体「日本システム・ダイナミクス学会」（これは国際学術団体の System Dynamics Society の日本支部でもある）は「システム・ダイナミクス」の表記を用いている。そこで本稿はこの表記にしたがう。

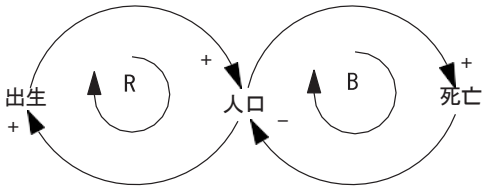
なお、システム・ダイナミクスの専門用語の定義については Ford（2019）にまとめられている。

２．システム・ダイナミクスの特徴

システム・ダイナミクスは他のアプローチと比較した際に以下のような特徴を持っている。

- フィードバックループ構造（因果関係の循環構造）に注目
- 描き方・読み方の決まっている図（因果ループ図・ストックフロー図）でモデルを表現（図１と図２参照）
- 変数をストック（状態変数）とフロー（ストックの変化）、補助変数（他の変数からただちに計算される変数、外生変数、定数）に分類
- 非線形な関係の積極的な利用

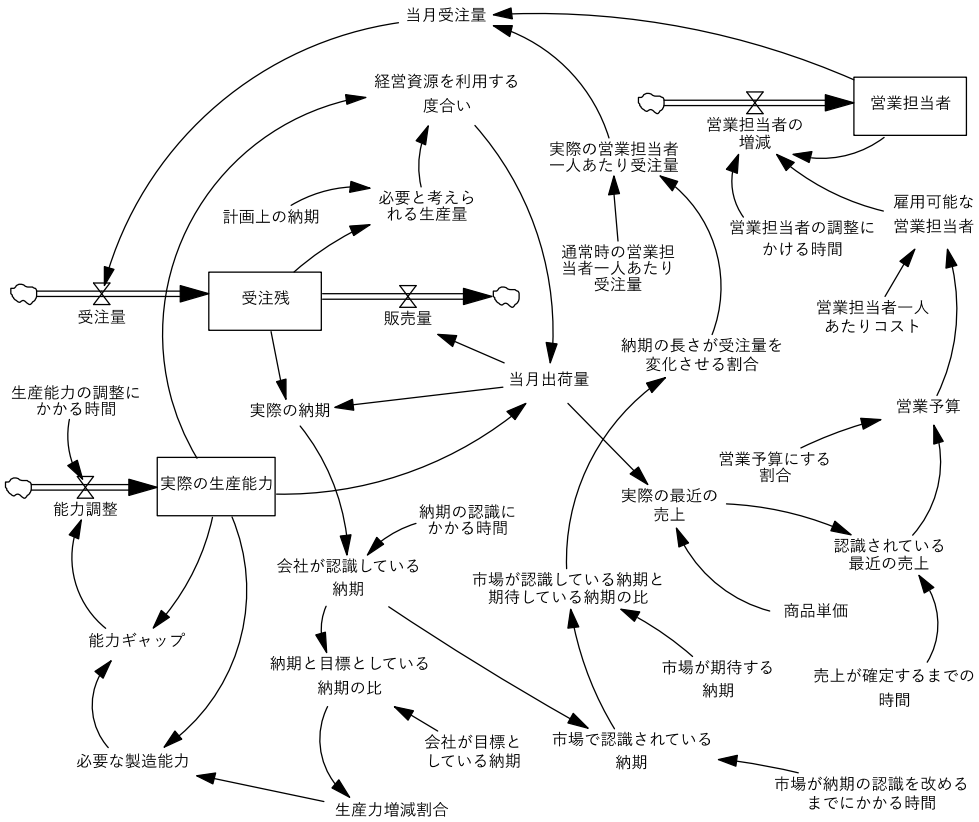
図１　因果ループ図の例。人口の増減を引き起こす構造を描いたもの



システムの挙動は、各時点におけるシステム内に存在する各フィードバックループの相対的な影響力の強さにより決まる。フィードバックループ構造は、情報参照（因果関係）の循環である。その相対的な影響力は時々刻々と変化する可能性があり、その変化は非線形な関係によってもたらされることが多い。システム・ダイナミクスは逐次的に計算するシミュレーションを利用するため、式の変形によりシステムの特徴を見いだす解析的なアプローチとは異なり、変数間の関係を無理に（例えば線形に）単純化する必要はない。なお、フィードバックループの相対的な影響力の計算方法として Loops that Matter method

（LTM）がある（Schoenberg et al., 2020）。ソフトウェア Stella Architect（本稿執筆時点のバージョン）は LTM を実装しており、自動的に計算・可視化できる。

図２　ストックフロー図の例。Forrester（1968a）を Sterman（2000, Chapter 15）で現代的な表現で表したものをもとに筆者が作成したもの。



ヒトや社会の反応はしばしば非線形であり、これを線形に単純化すると現実に起きている事象がシミュレーションでは再現できないことがあり得る。実際、社会において企業も個人も「想定外の顛末」に驚き、対応に苦慮することがよくある。Forrester（1971）はこれを「直感に反するふるまい」と呼び、フィードバックループ構造と非線形性にその根拠を求めた。Meadows（1982）はこれが実社会における介入の失敗など、「合理的に考えた末の行動がうまくいかない」ことにつながると指摘している。

システム・ダイナミクスではシミュレーション時間を連続的に捉えている。もちろん、コンピュータシミュレーションを実行する際にはソフトウェアにより離散化を行うが、モデリングする際は時間が連続的に流れている前提で行う。また、各変数がとる値も連続値であり、現実社会において離散値であるはずのものについては四捨五入などで整数にまるめて解釈する。この発想により、システム・ダイナミクスはシステム全体のふるまいを生

み出すメカニズムを理解し、改善・課題解決の要諦を見いだすことに適している。システムを構成する個々の要素が正確に各時点でどうなっているのか、たとえば、「たくさんのバス路線にたくさんのバスが走っているときに、どのバスがどの停留所にいるのか」を個別にシミュレーションするのは、システム・ダイナミクスを使うことが不可能ではないにしても他の手法の方が向いていると思われる。これに対して、「バス会社の持っている車両資源、人的資源、燃料補給拠点、その他様々なリソースのどの利用ルールをどう変更すればより良くなるか」を、システム内の要素を全体として俯瞰して捉えた値をパラメータとして与えて、フィードバックループ構造を根拠に「なぜそうなるか」を説明しやすい形で探っていくことにシステム・ダイナミクスは向いている。つまり、システム・ダイナミクスはシステム内の個別の要素に対する操作を最適化するために用いるということよりも、システム全体を俯瞰した課題認識・課題解決のために使うと効果的である。

他の細かい特徴については高橋・田中（2017）に説明がある。また、稗方・高橋（2019）ではシステム思考のアプローチ・手法のビジネス応用に関して概説している。システム・ダイナミクス自体の説明とビジネス面での応用は Sterman（2000）と Warren（2008）が、経済学との関連や発展的話題については Cavana et al.（2021）が、環境関連のモデルに関しては Ford（2009）が、公衆衛生関連については Homer（2012）と Homer（2017）が、モデリングとシミュレーション分析について書籍にまとめている。

３．モデル構築とシミュレーション分析の流れ

システム・ダイナミクスの一般的な分析の流れは様々な文献で示されているが、基本的には同一である（Martinez-Moyano and Richardson, 2013）。広く教科書として利用されている Sterman（2000, p. 87）をもとにすると、はじめに「課題の明確化」（リファレンスモードの設定）、次に「ダイナミック仮説の構築」（ここまでを概念化と呼ぶことがある）、「定式化」、「シミュレーションの実行とテスト」、「シミュレーション分析の評価と施策決定」、そして状況が更新された後に新たな「課題の明確化」（はじめに戻る）という流れである。ただし、このプロセスは一方向に進むとは限らず、必要に応じて戻ることはあり得る[2]。

モデリングの助けとなる考え方はこれまでいくつか紹介されている。Sterman（2000, pp. 79-81）は 12 箇条の原則を示している。このうちの初めに挙げられている、モデル構築の目的について「課題を解決するためにモデルを開発する」のであり、「『システムをモデル化する』こと自体を目的としない」という点は特に重要である。これは必ずしも、シミュレーションモデルは個別のケーススタディにのみ有効ということを意味しない。課題発生の理由やそれに対するリアクションパターンは、個々のケースや分析対象の枠組みを超えて一般性がありえる。モデル構築者は課題に向き合っている組織の構造のみに目を配るの

[2] むしろ、「済ませたプロセスをやり直したくない」という誘惑に打ち勝つことが重要である（Randers, 1980, p. 137）。

ではなく、課題を生み出す環境や意思決定ルールにも関心を向けることにより、普遍的知見を得られるモデルを構築できる。

ほかに、Warren（2014a）はシミュレーション可能なモデルを現実のふるまいを参照しながら早い段階から開発していく方法を示している。また Warren（2014b）は業種に依らず企業が持つと考えられる基本構造を示している。Takahashi（2008）はモデル化対象を説明する自然言語の文章を中間言語文に変形した上でモデル構造と対応づける方法を示している。

以下では、上で挙げた Sterman の示すモデリングのプロセスのそれぞれに関する説明を記載する。

3.1 課題の明確化

モデル構築者はモデルを使って何を検討・解決したいのかをはじめに明確化する必要がある。「解決すべき課題は誰にとってどんなものであるか」「それは何の量・水準をもって課題の残されている状態・解決された状態を表すのか」を、利害関係者や関係する従事者に聞き取り、明確化する。この情報はモデル化の範囲や細かさを定める根拠となる。このステップのヒントとなるものとして Ford and Sterman（1998）が暗黙知をいかにして形式知として得て、モデルの改善に役立てるかについて説明している。

このステップでは最終的に、課題の状態（未解決・解決を判断できる指標）やその他関連・関心のある情報について、過去から近未来についてどのように推移するかを整理する。過去のデータがない場合は今後の推移の予想のみで良い。これを時系列の折れ線グラフで描き、関係者の合意をとりまとめる。このグラフを「リファレンスモード」または「BoT（Behavior over Time）グラフ」と呼ぶ。リファレンスモードは手書きのラフスケッチであっても作る方が良い。これを見ることで、関係者が課題発生周りの事柄を、「どのくらいの期間にわたって見ているのか」ないし「どのくらいの時間の細かさで推移に注目しているのか」を知ることができ、より適切なモデリングにつながるためである。なお、シミュレーションの最初のマイルストーンとしてリファレンスモードの再現がなされるべきである。

Richardson and Pugh III（1981, p. 22）は「リファレンスモードを作成することを省略して成功した SD の利用例はほとんどない」と述べている。これは、物理・工学のシミュレーションに比べると、社会システムのシミュレーション分析では「どのような挙動が起きていたか・起きるはずか」という認識について、当事者間で差異が大きい場合があることが一因と考えられる。

また、リファレンスモードはモデルがどの程度広い範囲までカバーするべきか、およびどこまで細かく作り込むかの基準になる。リファレンスモード無しにモデル構築をはじめると、際限なく広く細かいモデリングになりがちである。Box（1976）の「全てのモデルは誤っている（一方で、便利なものもある）」という指摘の通り、モデルとはなにかの目的のために作られるものである。したがって、モデルはモデル化の対象そのものと同じではなく、その作成目的のために何らかの抽象化や省略がなされる。また、細かく作り込んで現

実のシステムの限りなく細かい写し取りをすることは、作業の終わりが実質的になくなり、そして、現実と同じだけの複雑さを持つことになり、得られる知見の質や分析結果の説明力が損なわれる可能性がある。

このような過度に詳細な作り込みを行おうとする可能性があるのはモデル構築者だけではない。Roberts（1977）は「クライアントはしばしば必要なレベルの 2 倍以上の詳細さを要求する」と指摘している。これを引用しつつ Sterman（2000, p. 217）は Robert の指摘を「過小評価（現実にはもっと細かく作り込むことを要求するクライアントがいる）」と述べている。モデル化において念頭に置くべきなのは「課題」であり、現実に存在するシステムの詳細構造の写し取りではない（Richardson and Pugh III, 1981, p.18）。

3.2 ダイナミック仮説の構築

ダイナミック仮説は、モデルに含まれる変数間にどのような関係があるか、またどのようなフィードバックループ構造が含まれているかについての仮説をまとめたものである。

ダイナミック仮説は通常、因果ループ図の形式で書かれることが多い（図 1 参照）。因果ループ図では変数間の因果関係を矢印で示し、値の変化の特徴から矢印に符号を付す。矢印は、ある変数 A の計算の根拠となる変数 B（原因）から、計算する変数 A（結果）にむけて引く。それぞれの矢印は以下のルールで正負の符号を付す（Richardson, 1986）。

- 原因の変数がより大きい場合、そうでなかったときと比べて結果の変数がより大きければ ＋
- 原因の変数がより小さい場合、そうでなかったときと比べて結果の変数がより小さければ ＋
- 原因の変数がより大きい場合、そうでなかったときと比べて結果の変数がより小さければ －
- 原因の変数がより小さい場合、そうでなかったときと比べて結果の変数がより大きければ －

ここでは一点注意が必要である。たとえば、出生数と人口は同じ方向に変化するようにも思えるかもしれない。しかし、出生数が仮に少なかったとしても出生数は人口を増やす「増加量」であるため、やはり人口は増加する。したがって、変化が同じ方向・逆方向という整理の仕方は不適切であり、いくつかの文献で使われた「s（same）」「o（opposite）」という記号を正負の符号に替えることは適切ではない（Richardson, 1997）。

矢印の循環が「フィードバックループ」（あるいは単に「ループ」）である。ループ内のある変数の変化が、その循環を経て将来再び同じ変数に同じ方向の変化を起こす場合は「自己強化型（reinforcing）ループ」、同じ変数に逆方向の変化を起こす場合は「バランス型（balancing）ループ」と呼ばれる。より簡単には、ループを構成する負の矢印の数で判定

できる。

・ループ内に偶数個（0 も含む）の － の矢印があれば自己強化型ループ
・ループ内に奇数個の － の矢印があればバランス型ループ

自己強化型ループは「R」の文字を矢印で囲った記号を図中に示し、バランス型ループは「B」の文字を矢印で囲った記号を示すことが一般的である（図 1 参照）。

もしモデルに単一のループしかなかった場合、自己強化型ループであればそれに含まれる変数は発散する。一方でバランス型ループであれば、収束、または振動する。バランス型ループが振動するのは、ループ内に 2 つ以上のストック（次節のストックフロー図の説明で述べる）がある場合のみである（Forrester, 1968b, pp. 10-15－10-20）。

現実には複数のフィードバックループが一つのシステムに含まれていることが通常である。したがって、上述の「単一のループしかなかった場合」のような挙動がはっきりと永続的に現れることは、少なくとも現実課題を解決するためのモデルではまれである（教育用の単純なモデルではあり得る）。現実には、各時点で相対的な影響力の特に強いループの特徴的なふるまいが現れる。

描かれるべきダイナミック仮説は、「ステークホルダーの関心のある変数」が含まれ、「リファレンスモードが再現される可能性があるフィードバックループ構造」（すなわち、上述のループの生み出す典型的なふるまいの組合せでリファレンスモードで表されたふるまいが起き得る）が含まれる最小構造（あるいはそれを含む必要最小限で拡張したもの）を描いた因果ループ図である（Randers, 1980, p. 131）。

ダイナミック仮説自体はモデル完成の後にシミュレーション結果から「妥当であろう」と判断できるとしても、論理的に「正しい」ということを証明することはできないので、分析が終わった段階でも「仮説」のままである。この仮説は何度でも検証され、更新される可能性がある。また、この後に作るストックフロー図への機械的変換方法はない（逆方向は可能）。したがって、いたずらに精緻で複雑な構造のダイナミック仮説を作ることに労力をかける必要はない。ステークホルダーが納得いく因果関係が描ければ、ダイナミック仮説としては十分である。

モデル構築者やステークホルダーの理解促進や、より平易な表現のため、後に述べるストックフロー図を混ぜた表現（ハイブリッド因果ループ図）にしている例もある。

3.3　定式化

定式化のプロセスはシミュレーションを行うにあたっての前提と言える「モデルの基本設定」と、「個別の変数の設定」が含まれる。ここではこれらの説明にあわせ、システム・ダイナミクスでしばしば使われる構造や定式化における総合的な注意点についても触れる。

3.3.1 モデルの基本設定

はじめに行われるべきことは、シミュレーションを「いつから開始して、いつ終わるか」、「どんな時間の単位でシミュレーションを進めるか」、および「計算の細かさをどの程度にするか」を決めることである。シミュレーションは単純に0時点目（0ヶ月目、0年目）から開始でも構わないし、暦年にあわせて2020年開始、とすることもできる。終了時点も同様である。

モデルの時間軸の1単位は月なのか、年なのかもはじめに十分検討すべきである。後から変更は可能であるが、問題の認識の仕方と関連しているため、はじめに十分に時間をかけて検討すべきである。一般的には、変化を見たい最小単位と同じか、それより細かい単位にするべきである。すなわち、月ごとの変化を見たいにもかかわらず年単位のシミュレーションをするのは不適切ということである。

なお、Ford（2018）は異なるタイムスケール間に存在する依存関係を検討する必要があるとき、それに対応する方法を示している。

計算の細かさ（DT、あるいはTIME STEPと呼ばれることが多い）は、原則として、より小さなDTにしてもシミュレーション結果が変わらない程度に小さくする必要がある。正確には、モデルに含まれる「時間の要素（変化の平均時間や輸送時間等）τ_i」および「遅れの時定数d_iをその遅れの次数o_iで除した値（後述）」のうち最も小さな値の半分未満である必要がある（式(1)を参照）。

$$\mathrm{DT} < \frac{\min_i\{\tau_i, d_i/o_i\}}{2} \tag{1}$$

これは、システム・ダイナミクスのシミュレーションソフトウェアの多くが採用している数値計算アルゴリズム（オイラー法等）による制約である。これに関する数学的な説明はSterman（2000, pp. 907-910）にある。

3.3.2 個別の変数の設定

システム・ダイナミクスの定式化は、ストックフロー図の作成と各変数の値や式の定義でなされる。

各変数はストック・フロー・補助変数に分類される。ストックはある時点にたまっている量、残っている量、水準である。たとえば、バスタブ内の水量、倉庫の品物の数、各時点の電力消費水準などである。ストックにはシミュレーション開始時点の値（初期値）を与える。

フローは各ストックの変化（流入・流出）を表す。たとえば、ストックとして人口があれば、フローとして出生・死亡が考えられる。フローの式としては接続されているストックの単位時間あたり増加量・減少量を与える。以下にRichardson and Pugh III（1981, pp. 134-

154）で挙げているフローの定式化の基本形の例を挙げる。

・フロー ＝ ストック × 単位時間あたりの変化率
・（流出フローのみ）フロー ＝ 流出元ストック ÷ 流出元ストックでの平均滞留時間
・（流入フローのみ）フロー ＝（目標値 − 流入先ストック）÷ 平均調整時間
・フロー ＝ 通常時の変化量 ＋ なにかの影響による補正値
・フロー ＝ 通常時の変化量 × なにかの影響倍率

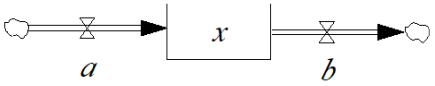
なお、上記の式中の変化率、平均滞留時間、平均調整時間、補正値、倍率は定数とは限らない。むしろ他の変数により変化する可能性があるという前提でモデルの構造を探ることが重要である。また、流出フローは因果ループ図と矢印の向きが逆になることに注意する。フローは常にストックの変化の原因となるものであり、図上の矢印の向きに違いがありながらも因果関係に変化はない。

補助変数は他の変数からただちに計算される量、外生変数、定数[3]である。「ただちに計算される」とは、他のモデル内の変数を加減乗除または何らかの関数で変換した後に得られるという意味である。たとえば、ある国の人口のモデルを作るときに「女性の人口」「男性の人口」という二つのストックがあるときに、これらを合計することで「総人口」という変数を作る場合、総人口はストックとしての特徴を持っているものの、他の既存の変数を足し合わせたものとしてただちに得られるため補助変数となる。

多くのソフトウェアはこれらの各変数の種類毎にほぼ同じ見た目のアイコンを定めている。ストックは長方形に変数名、フローは二重線矢印または太線矢印に変数名、補助変数は円形に変数名かアイコン無しで変数名のみである。各変数の間の因果関係（計算に使う情報の提供元と計算される変数の関係）は、細い矢印で表現される（図２参照）。なお、フローからそれが変化させるストックに向けては細い矢印を引かない。

ストックは微分方程式における状態変数、フローはその微分の構成要素に対応する。すなわち、システム・ダイナミクスモデルは実質的に連立微分方程式モデルと同じものと言える。なお、実際に定式化のプロセスで与える式は、むしろ積分方程式の形に近い。図３に示したストックフロー図は、式(2)に対応する。

図３　単純なストックフロー図の例。



[3] 定数を変数と呼ぶのは妙ではあるが、伝統的にこう呼ばれている。補助変数という「表現形式」であるとも言える。なお、英語では auxiliary または converter と呼ばれることが多い。

$$x = \int (a - b)\,\mathrm{dt} \tag{2}$$

ここで、３つの基本的注意点を挙げる。一つ目は「ストックは直接接続されたフロー以外で変化することはない」という点である。たとえばバスタブの水量は一瞬で望みの量に変化することはなく、一定時間をかけて流入流出をさせなければ変化しない。つまり、他の変数から因果関係を表す細い矢印が入ってくることはない[4]。

二つ目は「フローは他の変数の計算根拠にならない（つまり細い矢印を引き出す元になれない）」という点である。コンピュータソフトウェアの近似計算の中でフローはいくつであるかを表示可能であるが、現実の世界では各瞬間に正しく認識されることはない。認識できない以上、それを根拠として計算される量はない。我々が日常で「変化量」「変化率」を根拠に意思決定している場合、それは過去の（移動）平均をとっているなど、何らかの処理をされた情報である。システム・ダイナミクスではこの処理済みの値と実際の値を区別して表現する。

三つ目は「全ての変数には適切な単位を定義し、変数を一つ追加する毎に矛盾がないかを確認する」という点である。多くのソフトウェアがこの単位確認の機能を持っている。なお、フローは接続されているストックの単位をモデルで使用している時間の単位で除したものでなければならない。なお、単位の整合性がとれていても「そのモデルは正しい」ということはできない。しかし、単位の不整合があるモデルは明らかに誤っている。因果ループ図とは異なり、ストックフロー図ではこのような整合性を徹底的にとって作成していくため、概念化段階で単純な因果ループ図だったものが複雑なストックフロー図になることは十分にあり得る。しかし、シミュレーション研究ではモデルの合理的説明ができなければ、モデルを妥当なものと見なすことは困難である。モデルが複雑化した後で単位の整合性を確認するのは大変な労力が必要になるので、モデルの作成中に随時確認すべきである。

一般的には、はじめにシステム全体の変数が均衡状態になるようにパラメータを与え、これにショックを与えるなどして適切に変化することを確認しながらモデルを作っていく。なお、前述のとおり Warren（2014a）は実際の値を与えつつモデルを作っていく方法を提示している。どちらの方法も適材適所で使われるべきである。

なお、ストックフロー図の全体像を先に作ってから値や式を与えていくことは、モデルの完成を困難にする場合がある。この手順では、シミュレーション結果が明らかに不適切な挙動となった場合にどこに誤りがあるのかを特定することは困難であり、全ての変数を確認することになってしまう。まず容易にふるまいが予想できる小さな構造に値を仮置きしつつ動作を確認し、その後にそれらを少数組み合わせてふるまいを確認し、最終的に必要な規模のモデルにする手順のほうが、結果的に手戻りが少なく効率よくモデルを作成で

[4] Vensim など、ストックの初期値を与えるために矢印を引くソフトウェアはある。

きる。

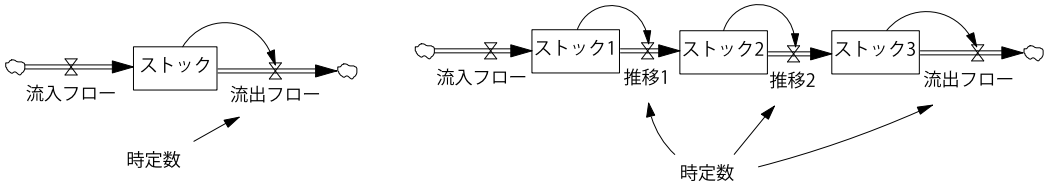
3.3.3 よく使われる構造

遅れ　現実社会ではものの移動には時間がかかり、状況の認識（受容）には時間がかかる。システム・ダイナミクスではこれら時間のかかる現象を「遅れ」と呼んでいる。遅れは日常生活や業務では「悪いもの」というイメージを持ちやすいが、システム・ダイナミクスでは特に悪いニュアンスはなく、単に時間がかかることを指している。そして、この遅れが一見合理的に見える介入を無効化する、あるいは意図しない結果を引き起こすことがしばしばあるため、システム・ダイナミクスはモデルに遅れを積極的に反映させる。

遅れには「モノの遅れ」と「情報の遅れ」の 2 種類がある。

「モノの遅れ（material delay）」は物体の移動や状態遷移など、ある変化が次の変化の発生までに時間がかかることを表す構造である。流入フローで表された何らかの変化が流出フローで表される次の変化まで､間にあるストックに滞留するという概念である(図４参照)。

図４　モノの遅れの構造。左が１次遅れで右が３次遅れ。



２つのフローの間に挟まれたストックの個数が遅れの次数と呼ばれる。このストックフロー構造を流れる要素は多様である前提があり、ある要素は素早くストックを流出し、別の要素は長く滞留する。ストックフロー構造内を流れていく要素が間のストックに滞留する時間の平均を「遅れの時定数」と呼ぶ。各流出フローは流出元ストックの値を「時定数 ÷ ストックの個数」で除したものとして定式化される。実際のモデリングにおいて間のストックの値を必要としないモデルの場合、モノの遅れは各シミュレーションソフトウェアの持つ組み込み関数で表現することが多い。

遅れの次数はモデルにおける仮説（何段階を踏んで最終状態を脱するか）に基づいて決めるほか、データから得る方法もある。過去のデータから遅れのプロセスに入る要素（はじめの流入フローに到達する要素）が遅れのプロセスから出る（最後の流出フローに到達する）までの時間が得られている場合、遅れの次数oは個々の要素の流出までの時間の平均値$\bar{d}$の二乗を個々の要素の流出までの時間の分散sで除した値について小数点以下を丸めたもの（式(3)参照）で得られる（Sterman, 2000, pp. 465-466）。

$$o \approx \frac{\bar{d}^2}{s} \tag{3}$$

なお、間のストックの個数が無限大になったものが「パイプライン遅れ」と呼ばれるものである。パイプライン遅れでは、流入フローの値は指定した時間だけ遅れて流出フローの値となる。

遅れの時定数は遅れのプロセスを通過する全要素の平均滞留時間であるため、直感的に「多くの（あるいは半分の）要素がその時間に（あるいはその時間までに）遅れのプロセスを通過する」とイメージしがちであるが、実際は異なる。１次のモノの遅れを例にすると、例えば図４左の流入フローが０時点目で大きさ a のパルス入力であった場合、ストックの値は式(4)の x で与えられる。なお、遅れの時定数は d である。

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -\frac{x}{d} \tag{4}$$

これを解けば式(5)となる。

$$x = ae^{-\frac{t}{d}} \tag{5}$$

したがって、パルス入力で与えられた流入フローは時定数で与えた時間が経過したときにはすでに約 63%が遅れのプロセスを抜け出ており、ストックには約 $0.37a$ が残るのみである。

一方で、当事者の人々に「多くの要素がこのプロセスを終えるにはどのくらいの時間がかかるか」ということを尋ねると、自分の経験した半数程度がプロセスを終えた時間を答える可能性がある。これは上の例で言えば $x = 0.5a$ となる時点を答えていることになる。この時点は放射性物質の半減期に対応する。半減期は平均寿命（遅れの構造に含まれるストックへの平均滞留時間であり、遅れの時定数に対応する）よりも短いので、この情報をそのまま遅れの時定数とすると過少に見積もることになる。ヒアリングをベースに遅れの時定数を定める場合、このような過少推定をしてないか確認することが望ましい。半減期に相当する値を h とすると、上の例であれば式(5)から式(6)と表せる。

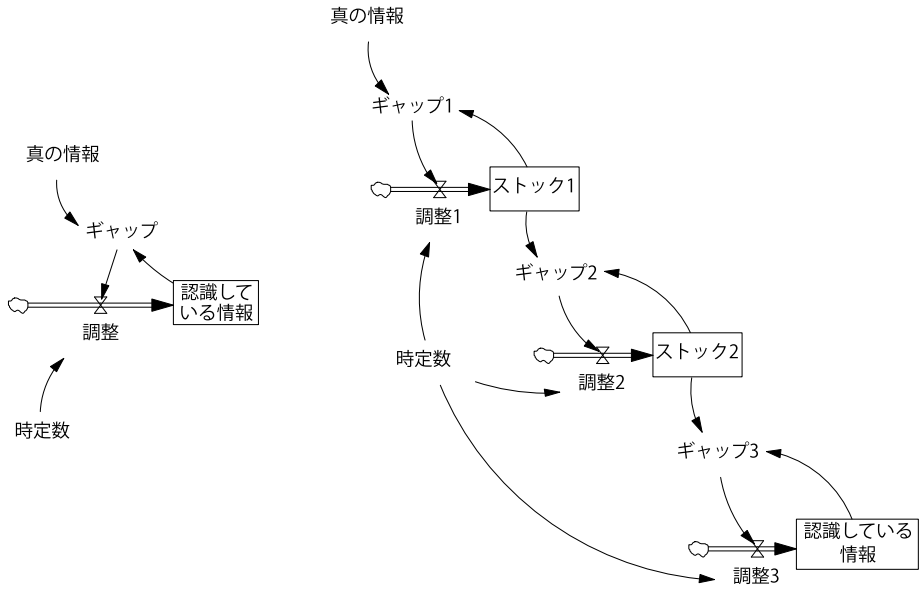
$$0.5a = ae^{-\frac{h}{d}} \tag{6}$$

これを解けば $d\ln 2 = h$ である。したがってヒアリング等で得ている値が半減期に相当する時間の可能性がある場合は、この値を $\ln 2$ で除した値（平均寿命）に変換して時定数にすると、実態に合うものとなる。

「情報の遅れ（information delay）」は、情報の認識・受容を表現する構造である。人や社会は、新しい情報や更新された（真の）情報を受け取ってもすぐにそれを受け容れないことがある。これは能力的な理由もあり得るが、「様子を見る」という行動が往々にしてリスク回避になるためである。たとえば、昨日までと異なり今日は急にたくさんの受注があっても、その瞬間に対応力を高めるべく人を雇う、あるいは工場のラインを増やすような意思決定は通常しない。その理由は、急な受注の増加は「例外」「突発的事象」で永続的な状態変化ではないかもしれないからである。このように、現実をすぐに受け容れずに時間を

かけて認識を更新していく様を表すことは、社会システムのシミュレーションでは大変重要である。情報の遅れもストックフロー構造で表現する（図５）。

図５　情報の遅れの構造。左が１次遅れで右が３次遅れ。



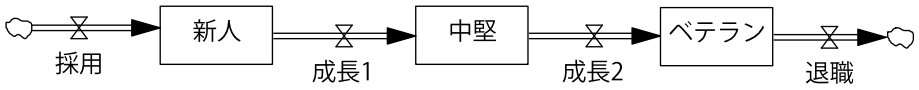
情報の遅れは認識されている情報をストックとして表し、これと「真の情報」とのギャップを、時間をかけて修正する「調整」というフローがあるという構造である。どれだけの時間をかけて修正するかは多様であるという前提（人により異なる、状況により異なるという意味）で、この平均値を遅れの時定数として与える。モノの遅れと同様に連結して高次数の遅れにすることができる。高次数の遅れは、情報の認識からその認識された情報の利用（意思決定）までに複数の階層が関与する場合に相当している。「調整」のフローは「真の情報」と「認識されている情報（ストック）」を「遅れの時定数 ÷ ストックの個数」で除したものとして定式化する。情報の遅れはフローの値を直接必要としない場合、多くのシミュレーションソフトウェアが実装している組み込み関数で表現できる。

また、情報の遅れは数学的には現時点に近い過去に大きな重みを与える指数移動平均に他ならない。過去の履歴にもとづく意思決定をする際にこれを利用する例も多い。なお、遅れの数理については Goodman（1974, pp. 219-239）と Sterman（2000, pp. 462-466）が詳細を説明している。

エージングチェーン（aging chain）　エージングチェーンは複数のストックを「ある状態にある要素の数」、それらをつなぐフローを「状態の遷移」として表したものである。たとえば、企業の従業員を見たとき、入社後時間の経過に伴い異なる状態になっていくとする。この場合、入社後の社員の状態を分けてモデル化することができる（図６）。これを発

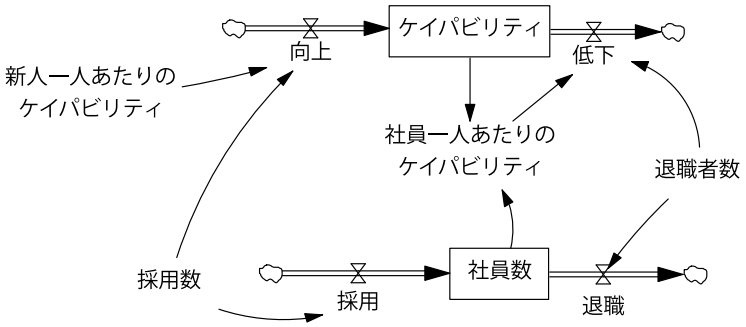
展させて、逆戻りのある構造などを作ることができる。一連のストックフロー構造から途中で漏れ出ていく構造もしばしば使われる。また、後述のコフローと組み合わせることも多い。

図６　エージングチェーンの構造。社員の状態遷移の例。遷移メカニズムは省略している。



コフロー（co-flow）　コフローは主に「要素数」とそれらの「特徴を表す量」とを区分して表現するものである。たとえば、社員数とその企業のケイパビリティを分けて考える例である。「100 人いれば 100 人力」とは限らず、「一部は新人、一部は中堅やベテランで、能力に差がある。新人はベテランや中堅に世話を焼いてもらうため、中堅やベテランも本人の能力を十分発揮できない。また、時間を追ってケイパビリティは向上したり、陳腐化するなどの変化がある」という場合、組織のケイパビリティは「人数」と「一人あたりに期待される能力」の単純な積とはならない。このギャップを適切に表現する場合、コフローと呼ばれる並行したストックフロー構造を用いることが多い（図７）。

図７　コフローの構造。組織のケイパビリティの例。詳細構造は省略している。



3.3.4　変数の定義における注意事項

テーブル関数（グラフ関数・ルックアップ関数）　システム・ダイナミクスでは非線形な関係を変数間に定義することが容易である。このとき、高次関数や指数関数等を用いることも可能であるが、ステークホルダーとの同意が取れるまでは、あえて入力と出力の関係を対応表（あるいはグラフ）で表すテーブル関数（またはグラフ関数、ルックアップ関数とも呼ばれる）を用いると便利である。この値の対応について同意が取れた後、高次関数や指数関数などを使って表現し直し、パラメータの最適化を行うことも可能である。

テーブル関数を作成する場合、その関数は単調増加または単調減少であるべきである（Richmond, 1992, p. 86）。そうでない場合、因果ループ図で表される２つ以上の因果関係

（矢印）を一つにまとめていることになり、モデルの可読性やテーブル関数として与えた変数の解釈が困難になる可能性が高いためである。Richardson and Pugh III（1981, pp. 173-174）はテーブル関数の定義についてガイドラインを示している。基本的には、テーブル関数の定義域は上限下限のうち、せめて片方は明確であるようにした方が良い。また、加算的に他の変数と用いる場合には「標準状態の時の値は０」、積算的に用いる場合には「標準状態の時の値は１」となるようにすると、ステークホルダーが理解しやすい。これを実現するため、テーブル関数への入力は（必要があれば出力も）、可能であれば標準化した値にする（単位を１となるように変換する）と扱いやすい。

数式の作成における表現　ソフトウェアのプログラミングと同様に、変数の定義式の中に直接定数[5]を使わないことは重要である。変数を「x×1.1」と定式化すると、乗じている 1.1 は何を意味するのかわかりづらく、また、この値が複数箇所で使われていたときにモデルのアップデート時に更新をし忘れる懸念がある。このような定数は補助変数として独立させ、モデルの中で同じ意味の定数は同じ補助変数を利用するようにすべきである。

もちろん、様々な定数を与えるために統計的手法を用いることは有効である。Senge（1977）や Graham（1980）はこれに関連した手法を比較検討している。その他の守るべき原則は Richardson and Pugh III（1981, pp. 261-266）に詳しく紹介されている。

3.4　シミュレーションの実行とテスト

モデルが完成した後は、シミュレーションを実行する。はじめに均衡状態を発生させるパラメータ設定をした場合はその確認と、パルス入力やステップ入力を確定数に与えた場合の変化を観察し、妥当であることを確認する。また、リファレンスモードが再現されることを確認する。

モデルのテストについては Forrester and Senge（1980）で多数の方法が上げられている。特に、定数や外生変数に極端に大きいあるいは小さい値を与えても、「もし本当にそんな値が与えられればシステムはこうふるまうはずだ」という結果になることを確認すべきである。モデル中の変数は（モデルの欠陥であってもなくても）モデル構築者が想定していない値を取っていることはあり得る。その際に現実には起きない反応をするモデルでは、現実の課題解決に役立つ方策を検証することはできない。Sterman（2000, p. 884）は「統計的および主観的な考察から予測される範囲の少なくとも２倍の幅でテストを行う」ことを推奨している。

これに加え、Sterman（2000, Chapter 21）は過去の値とのフィッティングで使うべき指標（たとえば決定係数ではなく MAPE（平均絶対パーセント誤差）などを使うべきことなど）や、Theil の指標からモデルの欠点の判断や改善の方策のヒントを得る方法を説明している。

モデルのテストは、モデルがある程度できあがってから行うことが散見される。しかし、

[5] このような数式中で直接与えられる定数は「マジックナンバー」と呼ばれることがある。

シミュレーション結果の品質担保のためには実質的にシミュレーションの実行と並行あるいは先立って行われるべきである。

また、モデル作成とシミュレーションの実行は、「示唆を得るために」行われるものである。もし、示唆が得られない、つまり当たり前のことや理論上起きるとされていることがシミュレーションで起きているだけで、改善策や新たな知見を得られない場合は、前のステップに戻ってモデル化の範囲の再検討や、本来存在するはずのフィードバックループや非線形な関係を見落としていないかを確認する必要がある。例えば以下に挙げる点を無視または見落としていることにより、示唆の得られないシミュレーション結果や現実に合わないシミュレーション結果になっている可能性がある。

・定数としていたものは実は定数ではなく、内部の環境により調整される変数である。
・暗黙のうちに無限に使えるとしていたリソースが実際には有限である。
・他の当事者が実は存在していて、様々なリソースを取り合っている。
・公式には明らかにされていない暗黙または秘密のルールがある。
・本来は原因となる変数群から得られる結果の情報であるにもかかわらず、その結果の変数を使って原因となる変数の値を計算している。

3.5　シミュレーション分析の評価と施策決定

3.5.1　シナリオ分析

モデルの検証を経て実用に耐えるモデルであることが確認された後は、様々なシナリオに対応した設定でシミュレーションを行い、取るべき施策の検討をすることが多い。このとき、それぞれのシナリオシミュレーション結果について、全ての変数が妥当な範囲、および物理的に適切な範囲で推移していることを確認する必要がある。たとえば 10 平方メートルの部屋に 1000 人の人を詰め込むことはできないし、一般的な自動車が 1 時間で 1000km 移動することはできない。検証段階でこのようなことが起きないことを確認済みであっても、施策の提案の際には万全を期して妥当な範囲に全ての変数の値域が収まっていることを確認すべきである。

3.5.2　ノイズ（乱数）

現実の世界の不確実な要素の再現や、それに対するシミュレーション結果の頑健性を検証するため、ノイズ（乱数）を与えてシミュレーション結果の検証を行うことも多い。このときは、ノイズの種類と周波数に注意が必要である。多くのシミュレーションソフトウェアでは複数の確率分布のノイズを用意している。社会システムの場合、自己相関があるノイズで表現されるべき要素もある（たとえば、株価や為替相場など）。このような場合は正規分布や一様分布ではなくピンクノイズを利用する必要がある。

また、ノイズの周波数（変化の頻度）が高すぎると、シミュレーション結果にあまり影

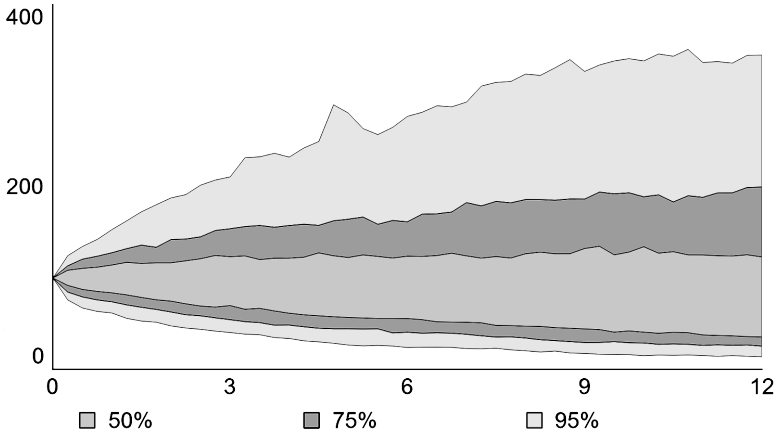
響を与えないこともあり得る。したがってノイズへの反応（シミュレーション結果の頑健性）を調べる際は、ノイズの周波数にも注意する必要がある。多くのシミュレーションソフトウェアでは、DT（TIME STEP）毎にノイズを更新するため、小さな時定数を設定したモデルの場合はノイズの周波数が過度に高くなっていないか確認する必要がある（Ford, 2009, p. 174）。

Sterman（2000, pp.923-924）はノイズをシミュレーションで使う際のガイドラインを示している。特に重要なのは、実際のシステムで変動し、その変動がフィードバックループの相対的影響力を変化させる可能性のある定数をノイズに置き換えてテストすることである。モデルの挙動にほとんど影響を与えないパラメータにノイズを含めても、得られる示唆は少ない。

3.5.3　感度分析（モンテカルロ・シミュレーション）

定数を乱数化して大量に繰り返しシミュレーション実行を行い、モデルのシミュレーション結果がどのような範囲に収まるかを調べることができる。これを感度分析（またはモンテカルロ・シミュレーション）と呼ぶ。多くのシミュレーションソフトウェアでは感度分析を実行すると、各変数のシミュレーション結果の中央パーセンタイルを面グラフで表す出力ができる（図８）。500 回以上のシミュレーションを行うと各パーセンタイルの範囲は、95%の安全率での許容区間（Tolerance Interval）でカバーする範囲と概ね一致する（Hahn and Meeker, 1991, p. 325）。

図８　感度分析で得られる面グラフの例。中央 50 パーセンタイル、75 パーセンタイル、95 パーセンタイルでエリアが塗り分けられている。



3.5.4　最適化

過去の値とのフィッティング（キャリブレーション）や施策の実行プランを立てる（オプティマイゼーション）ために、モデルのパラメータの最適化を行う場合がある。これら

は闇雲に行われるべきではなく、できるだけ限られたパラメータについて、現実に意味のある範囲の中での最適化を行うべきである。

複数のパラメータの最適化は、個別に行った場合と同時に行った場合とで結論が変わる可能性がある。また、最適化の開始の際に与えたパラメータの値によって結果が変わることもある。現実のシステムへの対応策を提案する際は、これらの条件を変えて結論が変化しないかどうかを十分に確認する必要がある。

3.5.5　報告対象（ステークホルダー）に対するモデルとシミュレーション結果の説明

十分に検証を経たモデルを使いシミュレーションで様々な知見を得た後は、報告対象（あるいはステイクホルダー）にモデルの表している内容とシミュレーション結果から得られた知見、モデルやシミュレーション結果の限界を説明する。

このとき、ストックフロー図は往々にして大規模・複雑になっている場合が多いため、大まかな、各時点で影響力の強いループを抽出した因果ループ図を用いて説明すると効果的である。

また、因果ループ図の中の変数名や施策の提案の際は、ステークホルダー自身の言葉で説明を行うべきである。決して省略された変数名やモデリング担当者の間で決めた簡易な言葉、あるいは数学の専門用語ではなく、ステークホルダーがその業務で使っている言葉を使い、また値や程度はどういうアクションを行うことに対応するかを例示して説明すると、ステークホルダーにとって役立つ知見を提供できる。

ステークホルダーのモデルに対する信頼や納得が不十分である場合、施策が決定されてもコミットメントが得られず、最終的にプロジェクトが失敗することがあり得る。ステークホルダーの信頼醸成と納得感を高め、また未発見のモデルの瑕疵を発見するために、ステークホルダーの求めに応じてモデルの値や構造を変更してシミュレーションを実行してみせることは意義がある。

また、シミュレーションの結果推奨される施策について「ステークホルダーに負担が生じる」、あるいは「成果が出るのに時間がかかる」ということがシミュレーションからわかる場合は、それについても説明することが施策の効果的な実施のために必要である。

４．ドキュメンテーション

モデル全体や個別の数式・定数に関するドキュメントを作成することは重要である。ここで言うドキュメントとは「数式の意味、出典、簡略化・何らかの工夫をした場合の根拠」の記録である。ドキュメンテーションはモデル構築者自身が将来にわたって責任のあるアップデートを行うために必要である。また、他者が読むことを助け、提案や建設的な批判を行うためにも必要である。Richardson and Pugh III（1981, p. 219）は、ドキュメントのないモデルは「知的なゴミ」であると批判し、Sterman（2000, p.855）もドキュメントが無く

ては「科学的でも有用でもない」と述べている。

現在使用されているシミュレーションソフトウェアの多くは、ドキュメントをモデル構築と同時に記録していくことが容易に可能である。モデル構築が一段落した後ではなく、変数一つ一つの定義・更新の度に、ドキュメントを残すことが重要である。更新の際は「なぜ、元々の定義を新たに更新したのか」の理由も残すと、その後のアップデートや再利用で有用である。

５．おわりに

本稿ではシステム・ダイナミクスの概要とモデリングの要諦について説明した。システム・ダイナミクスが考案されて半世紀以上が経過している。この間、欧米を中心に方法論やモデリング、そして応用面の課題を次々と克服して使いやすいアプローチとなった。また、使いやすいシミュレーションソフトウェアも多数生まれた。

さらに、今日ではデータサイエンス・AI の手法も数多く実用に供される段階になっている。こうした手法に与えるデータとしてシミュレーション結果を使うことにより、モデルのより深い理解や想定していなかった介入施策などを発案することも可能になってくると思われる。

こうした環境変化から、システム・ダイナミクスはいよいよ活発に利用されるべき時期が来ていると考えられる。

参考文献

高橋裕・田中伸英（2017），「システム・ダイナミックス～コンピュータを利用した問題分析とデザイン～」『日本経営数学会誌』第 37 巻，第 1-2 号，29–45 頁.

稗方和夫・高橋裕（2019），『システム思考がモノ・コトづくりを変える』日経 BP.

Box, G. E. P. (1976). “Science and Statistics.” *Journal of the American Statistical Association*, 71(356), 791–799, December.

Cavana, R. Y., B. C. Dangerfield, O. V. Pavlov, M. J. Radzicki, and I. D. Wheat (2021). *Feedback Economics: Economic Modeling with System Dynamics.* Springer.

Ford, A. (2009). *Modeling the environment 2nd Edition*. Island Press.

Ford, A. (2018). “Simulating systems with fast and slow dynamics: lessons from the electric power industry.” *System Dynamics Review*, 34(1/2), 222‐254.

Ford, D. N. (2019). “A system dynamics glossary.” *System Dynamics Review*, 35(4), 369‐379.

Ford, D. N. and J. Sterman (1998). “Expert knowledge elicitation for improving mental and formal models.” *System Dynamics Review*, 14(4), 309‐340.

Forrester, J. W. (1961). *Industrial Dynamics*. The MIT Press, Cambridge, Ma.

Forrester, J. W. (1968a). “Market Growth as Influenced by Capital Investment.” *Industrial Management Review* (now *the Sloan Management Review*), 9, 83‐105.

Forrester, J. W. (1968b). *Principles of Systems*. The MIT Press, Cambridge, Ma.

Forrester, J. W. (1969). *Urban Dynamics*. The MIT Press, Cambridge, Ma.

Forrester, J. W. (1971). “Counterintuitive Behavior of Social Systems.” *Technology Review*, 73, 52‐68.

Forrester, J. W. (1973). *World Dynamics*. The MIT Press, Cambridge, Ma.

Forrester, J. W. and P. M. Senge (1980). “Tests for Building Confidence in System Dynamics Models.” *TIMS Studies in Management Sciences*, 14, 209‐228.

Goodman, M. R. (1974). *Study Notes in System Dynamics*. Wright-Allen Press.

Graham, A. K. (1980). “Parameter Estimation in System Dynamics Modeling.” In *Elements of the System Dynamics Method*, edited by Randers, J. The MIT Press, Cambridge, Ma, 143‐161.

Hahn, G. J. and W. Q. Meeker (1991). *Statistical Intervals: A Guide for Practitioners and Researchers*. Wiley.

Homer, J. B. (2012). *Models That Matter: Selected Writings on System Dynamics 1985-2010*. Grapeseed Press.

Homer, J. B. (2017). *More Models That Matter: Selected Writings on System Dynamics 2011-2017*. Creatspace Independent Publishing Press.

Lane, D. C. and J. D. Sterman (2011). “Jay Wright Forrester.” In *Profiles in Operations Research: Pioneers and Innovators*, edited by Gass, S. and A. Assad. Springer, New York, 363‐386.

Martinez-Moyano, I. J. and George P. Richardson (2013). “Best practices in system dynamics modeling.” *System Dynamics Review*, 29(2), 102‐123.

Meadows, D. H. (1982). “Whole earth models and systems.” *CoEvolution Quarterly*, Summer, 98‐108.

Randers, J. (1980). “Guidelines for Model Conceptualization.” In *Elements of the System Dynamics Method*, edited by Randers, J. 117‐139, The MIT Press, Cambridge, Ma.

Richardson, G. P. (1986). “Problems in causal loop diagrams.” *System Dynamics Review*, 2(2), 158‐170.

Richardson, G. P. (1991). *Feedback thought in social science and systems theory*. University of Pennsylvania Press, Philadelphia.

Richardson, G. P. (1997). “Problems in causal loop diagrams revisited.” *System Dynamics Review*, 13(3), 247‐252.

Richardson, G. P. and A. Pugh III (1981). *Introduction to System Dynamics*. The MIT Press, Cambridge, Ma.

Richmond, B. (1992). *Introduction to Systems Thinking*. isee systems, inc.

Roberts, E. (1977) “Strategies for effective implementation of complex corporate models.” *Interfaces*, 7(5), 723‐732.

Sargent, R. G., P. F. Ruth, and T. J. Schriber (2017). “History of the Winter Simulation Conference: Renaissance Period (1975–1982).” In *Proceedings of the 2017 Winter Simulation Conference*, 50‐59.

Schoenberg, W., P. Davidsen, and R. Eberlein (2020). “Understanding model behavior using the Loops that Matter method.” *System Dynamics Review*, 36, 158‐190.

Senge, Peter M. (1977). “Statistical Estimation of Feedback Models.” *Simulation*, 28, 177‐184.

Sterman, D., John (2000). *Business Dynamics*. Irwin McGraw-Hill.

Takahashi, Y. (2008). “Dynamic simulation modelling using descriptive information in natural language.” *International Journal of Simulation and Process Modelling*, 4(3-4), 215‐222.

Wagner, H. M. (1969). *Principles of operations research, with applications to managerial decisions*. Prentice- Hall.

Warren, K. (2008). *Strategic Management Dynamics*. Wiley.

Warren, K. (2014a). “Agile SD: Fast, Effective, Reliable.” In *Proceedings of International System Dynamics Conference 2014*.

Warren, K. (2014b). “Standard Cases: Standard Structures: Standard Models.” In *Proceedings of International System Dynamics Conference 2014.*