「どんなものでも計算します」社会システム科学の一方法論
――「ゆるシミュ」へのお誘い――

江端智一　Tomoichi Ebata

1　はじめに

こんにちは．江端智一と申します．普段は総合電機メーカの主任研究員として働き，休日になると趣味の研究やシステム開発をして，その内容を技術系 Web メディアに寄稿する「週末研究員兼ライター」をやっています．

私は，私のパソコンで私のためだけのシミュレーションプログラムを作って動かし，その結果を見ることが大好きです．自分のパソコンであれば，自分勝手にメモリやハードディスク（リソース）を使い倒しても文句は言われませんし，近年のパソコンの性能は，前世紀のスパコンの性能を軽くりょうがするほどにもなってきています．特に近年のメモリの価格の低下によって，パソコンの中で仮想の人や建物や自動車（オブジェクト）を大量に作って，それらの全部を同時に動かすようなシミュレーションも可能となってきています．基本的には，その対象が‘もの’（有体物）でさえあれば，どんなものでも試すことができます．本稿では，様々な現実世界の現象を「説明」または「予測（推測）」する社会システム科学の一つの方法として，現実世界の膨大な数の人間や乗物を，そのまま仮想世界のオブジェクトに置き換えるシミュレーションの具体例を紹介します．

このたび，御縁があって，電子情報通信学会の方から寄稿の御依頼を頂きました．しかも「学問の応用に関する内容であれば何を書いてもよい」という太っ腹な御依頼でした．「渋谷を歩いていたらスカウトに声を掛けられた」と語るアイドルと同じくらい，うさん臭いと思われるかもしれませんが，本当です．

「電子情報通信学会，大丈夫かな？」と，一瞬心配したりしましが，こんなチャンスは，私がアイドルとしてスカウトされるよりもまれなチャンスと考え，脊髄反射的に「お引受けします」と返事をしました．私の連載がきっかけで御依頼頂いたのですから，連載の内容で書くのがよいと考えました．

そこで，今回は，「社会の時事問題を，『正論』，『正義』，『理想』，『モラル』，『公益』などを一切無視して，『数字』だけを使って把握してみよう」という試みで続けている連載（後述）で行ったシミュレーションの中でも，かなり“いい加減”で“アバウト”で“緩い”シミュレーション「ゆるシミュ」（筆者が命名）を，「社会システム科学の一方法」として御紹介したいと思います．

2　連載「Over the AI－AIの向こう側に」「世界を『数字』で回してみよう」とは？

現在，私は，「EE Times Japan」に寄稿させて頂いています．EE Times Japan とは，電気・電子関係のデバイスやツール，手法に関する技術情報や，業界の最新動向を提供する Web メディア（http://eetimes.jp/）です．私のブログを読んで頂いていた編集部の方からお声を掛けて頂いたのがきっかけとなっております．

2012 年から「『英語に愛されないエンジニア』のための新行動論」で連載を始めてから，「江端さんの DIY 奮闘記 EtherCAT でホームセキュリティシステムを作る」，「実録！ネット詐欺」などを経て，現在二つの連載を担当させて頂いております（表 1）．

今回は，現在連載中のこの二つの中から，特に反響が大きかった三つのコラムを選んで，その中で私が試みた「ゆるシミュ」を御紹介したいと思います．

表 1　現在担当中の連載

連載名	URL
世界を「数字」で回してみよう	http://eetimes.jp/ee/series/1534/
Over the AI ―― AI の向こう側に	http://eetimes.jp/ee/series/3761/

©電子情報通信学会 2019

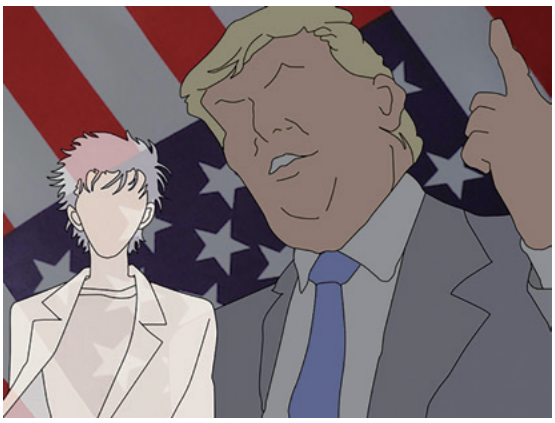

図１　連載イラスト「大統領選挙の結果にぼう然とする私」
（イラストも筆者が担当）

州	選挙人の数	勝った人		確保した選挙人	
		クリントン	トランプ	クリントン	トランプ
アラバマ州	9	×	○	0	9
アラスカ州	3	×	○	0	3
アリゾナ州	11	×	○	0	11
カリフォルニア州	55	○	×	55	0
・・・	・・・	・・・	・・・	・・・	・・・
合　計				232	306

図２　「選挙人制度」の概要

3　一つ目：米国大統領選挙のシミュレーション

このシミュレーションは，「何で投票数が少ない人物が当選するのだ？」という私の素朴な疑問から始まりました（図 1）．
(http://eetimes.jp/ee/articles/1611/29/news026.html)

きっかけは，御存じのとおり，2016 年の米国大統領選挙で，事前の世論調査を覆しドナルド・トランプさんが勝利したことです．しかも投票総数は，ヒラリー・クリントンさんが上回っていたにもかかわらずです．

もちろん私は，この結果が，あの国特有の「選挙人制度（その州で勝利した場合，その州の選挙人の全員が，勝利した候補者に投票する）」なるものによるものであることは，知ってはいたのですが（図 2），どの本や記事を読んでも，数学的，統計的な観点から説明をしている資料がなく，この制度を理解できず，気持ちが悪くて仕方ありませんでした．私は，どういう比率の投票がされれば，どのような結果になるのか（選挙人の数が決まるのか）を，「正確に」ではなくて「ざっくりと」知りたかっただけなのです．

「ならば，力ずくでシミュレーションしてやろう」と思い立ち，米国の各州の人口の有権者数と乱数関数だけを使った，メインルーチン 20 行という超簡単シミュレーションプログラムを 30 分ほどで作り，パソコンの中に約 3 億人の有権者（‘ひと’のオブジェクト）を作り，有権者が確率 50% でトランプさんとクリントンさんに投票するバーチャル選挙を 1 万回（4 万年分）ほど実施してみました．そして，その結果を見て，心底びっくりしたのです．

図 3 は，全ての有権者が（a）確率 50%:50% の投票を行った場合と，（b）49.99%:50.01% の投票を行った場合の，ある選挙人獲得数を得ることのできた（1 万回の中の）選挙の回数を示しています．

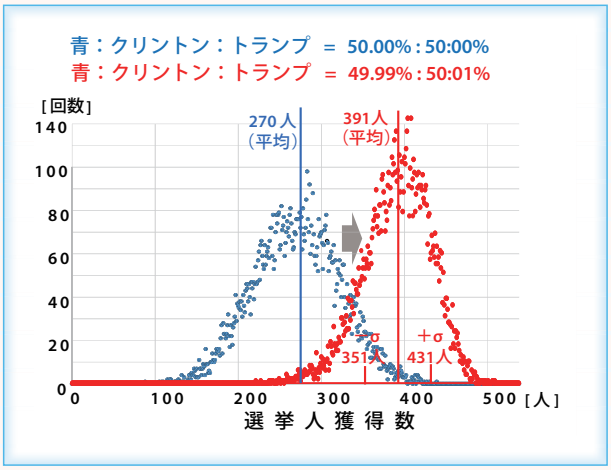

図３　大統領選挙 1 万回シミュレーション結果（1）

50%:50% であったとしても，選挙人獲得数が選挙をやるたびにこんなにも結果が変動する（分散が大きい）ことにも驚きましたし，たった 1 万人に 1 人（0.01% の有権者）の心変わりで，選挙人が平均 121 人増え，選挙の勝率が 99% 以上に跳ね上がってしまうという事実には，仰天しました（表 2）．「この選挙制度って，多数決で民意を反映することを目的としている選挙としては，世界最低じゃないかな」とまで思いました．

表 2　大統領選挙 1 万回シミュレーション結果（2）

(a)50.00% : 50.00% の時

候補者	勝ち数	勝率
クリントン	4855回	48.55%
トランプ	5076回	50.76%
引き分け	69回	

(b)49.99% : 50.01% の時

候補者	勝ち数	勝率
クリントン	36回	0.36%
トランプ	9961回	99.61%
引き分け	3回	

このシミュレーション結果は，ネットで大きな反響を呼び，ある協会から講師として招へいして頂いたり，シミュレーションの正しさを証明（リアプノフの中心極限定理で，正規分布への弱収束を証明）して頂いたり，更に定式化の方法を御教示して頂ける方も現われました．（御興味のある方は「抹殺する人工知能～生存競争と自然淘汰で，最適解にたどりつく（http://eetimes.jp/ee/articles/1701/31/news031_4.html）」を御一読下さい．）

メインルーチンたった 20 行の「ゆるシミュ」で，ここまで騒ぎになった（いろいろな人が興味を持った）ということは，“厳密”で“厳格”で“精密”なシミュレーションでなく，“いい加減”で“アバウト”で“緩い”シミュレーションであっても，社会のシステムをざっくりと理解するには十分に役に立つということです．

十分な時間を掛けて仮説検証やフィールドテストをすることも大切ですが，「30 分間程度のプログラミングでちょいちょいと試してみる」という軽い‘のり’も，これからの社会システムの研究には，大切だと思うのです．

4　二つ目：残業発生シミュレーション

このシミュレーションは，「時短（労働時間短縮）が達成できない？そんなもん，上司がいつまでもオフィスに残っているからだろうが」という私の考えを，私自身で納得するために始めました（図 4）．（http://eetimes.jp/ee/articles/1707/13/news016.html）

これは，現在連載中の「働き方改革」の第 1 回目のシミュレーションです．私は，昔から時短推進月間などでスローガンの募集があるたびに，“「時短」「時短」とうるさいぞ．お前（上司）が帰れば皆（みな）帰る”という川柳（？）を，何度も会社に投稿してきたのですが，過去に一度も採用されたことがありません．そこで「今度こそ，納得してもらおうじゃないか」と思い立ち，シミュレーションを試みました．

表 3 の仮説を一言で言えば「私のボーナスの査定をする上司がオフィスに残っていると，私は帰宅しにくい」を四つのユースケースで具体化したものです．その上で，パソコンの中に仮想の社員を作り，簡単なルールベース推論（ファジー推論）を用いて，その社員の意思決定アルゴリズムを組み込みました（表 4）．これは，自分の席と上司の席までの距離と，帰宅を思いとどまる時間を 15 のパターンに分類したルールとなっています．

図 4　連載イラスト「残業で疲れ果てた私たち」

表 3　仮説「上司が早く帰宅すれば全員が早く帰宅できる」

	仮説のポイント	シミュレーションの方針
# 1	自分の仕事が終わるまでは、帰らない	自分の仕事については、回りからの影響は受けない
# 2	自分の「一個」上の上司が残っていると、帰宅しにくい」	(A) 課長は部長を、(ヒラ) 社員は課長を気にするが、その逆は成立しない
		(B) しかし、(ヒラ) は「二個」上の部長は気にしない
# 3	上司の近くは「帰宅しにくい」	上司との距離 (例：席の近く) に、影響を受ける
# 4	「帰りにくい」といっても限度がある	超長時間 (例；2 時間) もダラダラと残業することはない

表 4　社員の意思決定テーブル（ファジールール）

		私の直上の上司の席から (私の席まで)				
		凄く近い	近い	そこそこ	遠い	凄く遠い
自分の仕事の完了後	直後	帰れない	帰れない	帰れない	迷う	帰る
	1 時間くらい	帰れない	帰れない	迷う	帰る	帰る
	2 時間以上	帰れない	迷う	帰る	帰る	帰る

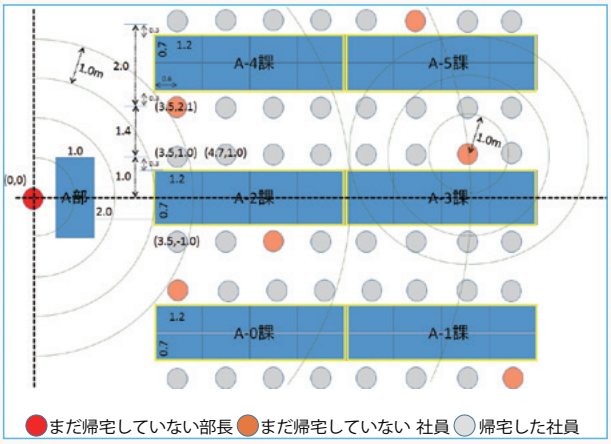

図５　想定したオフィスの環境

次に，仮想のオフィスの中に，部長，課長，平社員の仮想エージェント，合計 49 人分を作り（図 5），シミュレーションを実施しました．

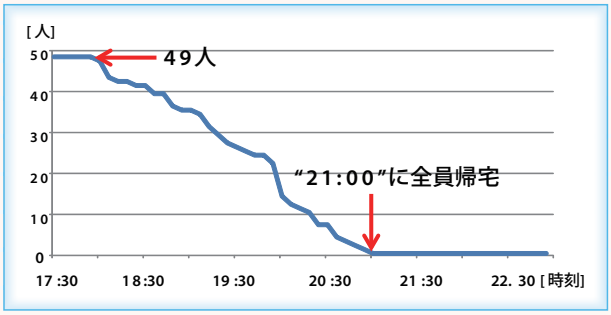

図６　部長がいない日のオフィスの残業人数の変化

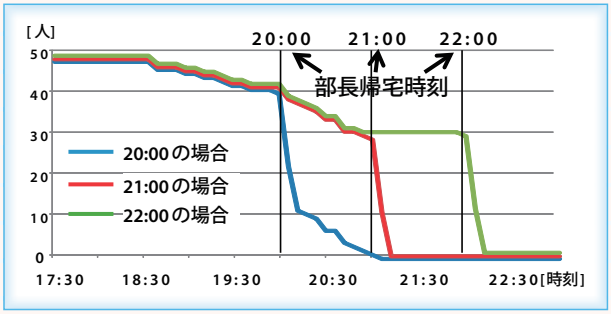

図７　部長が残業している日のオフィスの残業人数の変化

まず，部長がいない日のオフィスの残業人数の変化を示します．このように部長がオフィスにいない日は，部下たちが一定の比率で帰宅している様子がうかがえます（図 6）．

しかし，部長がオフィスに居座り続ける日は，部下たちは帰れなくなっていることが分かります（図 7）．部長の帰宅時間が遅れるほど，残業する社員も減っていかないことも分かります．このシミュレーションによって“部長という存在”が「残業」を助長することが分かりました．（私がそうなるようにプログラムを仕込んだのですが，この顕著な結果には，この私が驚きました．）

この「ゆるシミュ」で，「ざっくりとした，しかし，リアルなオフィスの残業の様子を見ることができた」と思っています．

私は，「世間を啓発するため」でもなく「正確な定量結果を得るため」でもなく，ただ，「この私が知りたいことを知るため」だけに，シミュレーションをしました．そして，それで十分であると思っています．「正しい仮説」とか「正確なオブジェクトモデル」とか，そんなことばかり考えていれば，私の人生は，何もしないまま終わってしまいます．

5　三つ目：江端専用ダイエットシミュレーション

このシミュレーションは，「痩せたければ，食うな」をきっちりと数字で証明してみせる，という私の執念から始まりました．世の中の「ダイエット本」と言われている内容のいい加減さに，心底頭にきていたからです．このシミュレーションでは，私の体を張った実証実験も行いました（図 8）．
(http://eetimes.jp/ee/articles/1505/14/news072.html)

その検証を行う前に，「どのようにダイエットは失敗

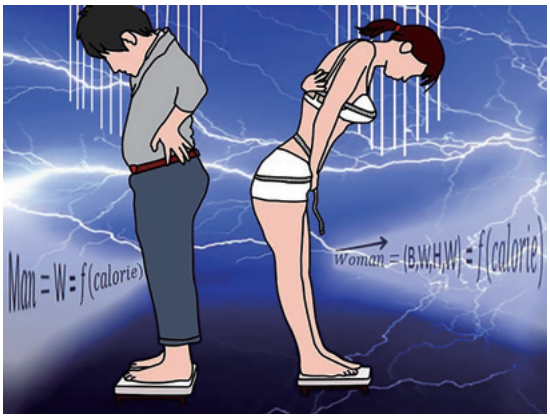

図８　連載イラスト「男女間のダイエットの観点の違い」

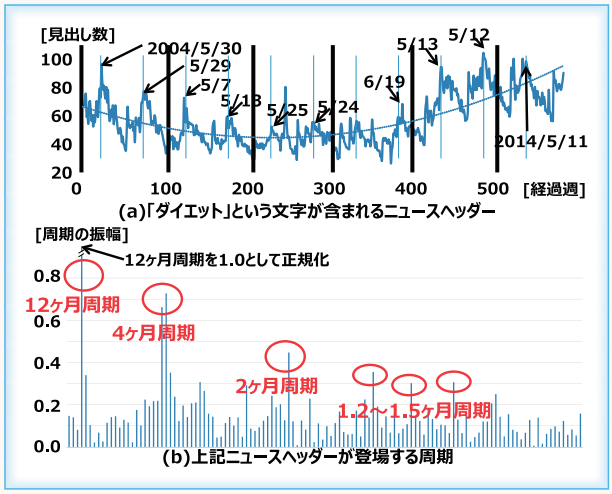

図９　「ダイエット」という文字が出てくるニュースヘッダの傾向

図 10　体を張ったシミュレーション

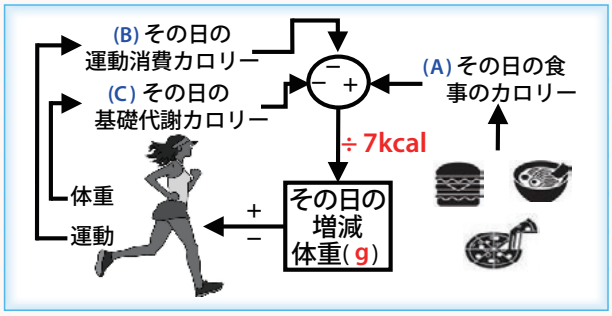

図 11　ダイエットシミュレーションの構成

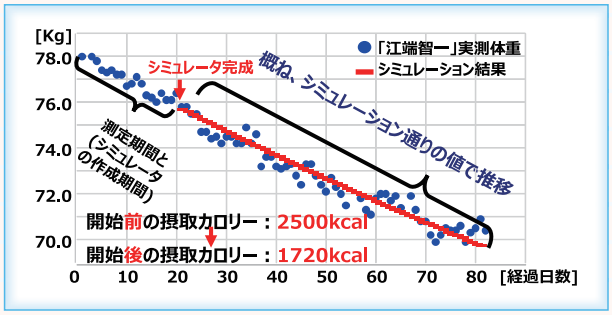

図 12　シミュレーション結果と実測値の比較

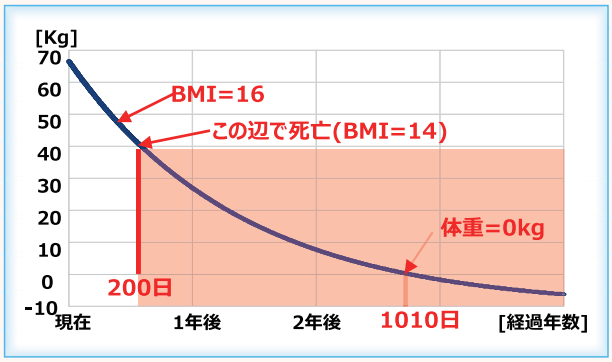

図 13　江端がダイエットで餓死する日の予測

していくのか」を調べてみました．具体的には，「ダイエット」という用語を含むニュースヘッダが登場する回数をフーリエ変換して，「ダイエット」という用語の出現頻度を調べてみました（図 9）．

この結果，４か月，２か月に強い周期性があることを発見しました．４か月周期は「新学期 / 入社式の４月」「水着の８月」「クリスマス（デート）/ 正月（太り）の 12 月」で理解できましたが，２か月周期の説明ができませんでした．このことから，私は「人間は一つのダイエットを，２か月以上続けるだけの根性がない」という仮説を導き出しました．

次に，私の「体を張ったシミュレーション」では，パソコンの中に，仮想オブジェクトとしての私（バーチャル江端）を作り（図 10），このバーチャル江端に対して，現実の江端と同じ，あるいは，無茶苦茶なダイエットをやらせて，どのような現象が発生するかを調べることを目的としたものです．そして，これも以下のモデルをプログラムした「ゆるシミュ」をやっただけです（図 11）．

ダイエット後約 20 日分の私の体重の変化から導き出した数値パラメータを使って，その後の体重とシミュレーション値を比較してみたら，ほぼ‘どんぴしゃ’で推移していることが判明しました（図 12）．人間というのは，この程度の簡単なモデル化ができる‘もの’であるということに，驚きました．ちなみに，私はこの１年間の連載中に，ちまたで騒がれているダイエット方式についても調べてみましたが，その結果として，私が読んだ何十冊のダイエット本の全部で，根拠となる理論も実験データも，１行も記載されていないことを知りました．

この連載の結論として，私は「痩せたければ食うな」という，「エネルギー保存法則だけが唯一のダイエットの真実である」と断定しました．（現時点でいかなる反論も抗議も受けていません．）

これ以外にも，この江端専用の体重シミュレータは，興味深いことをたくさん教えてくれました．例えば，図 13 は，私がこのダイエットを続ければ，200 日後に餓死することを示しています．また，このシミュレーショ

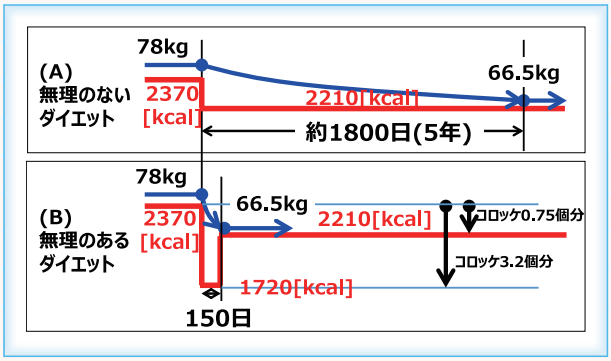

図 14　無理なダイエットを行う意義

ンは苦しいダイエットを実施する意義も教えてくれました（図 14）．いわゆる「楽々ダイエット」では 1,800 日掛かる減量が，「苦痛ダイエット」なら 150 日間だけ我慢すれば，その後は「楽々ダイエット」に移行しても体重は戻らないことなども分かってきました．

これが正しいことは，私がリバウンドすることなく現在も適正な体重を維持できていることからも，「裏」が取れていると思います．

つまり，「ゆるシミュ」であっても，その結果は，侮れないということです．また，「ゆるシミュ」を作るプロセスにおいて，私は，どういうオブジェクトを作らなければならないか，その上でどんなパラメータが必要であるか，どのような外部要素を組み込まなければならないかを，最低 1 回は真剣に考えなければなりませんでした．

その上で，「ゆるシミュ」をプログラミングすることは，その問題（社会システム）を考える上で，100 冊の本を読むよりも，はるかに深い理解を得ることになるのです．

6　スケーラブルリアルタイムシミュレーションを実現する手段

さて，ここまで紹介してきた三つのシミュレーションは，それほど多くのオブジェクトを必要としませんが，私の本業のシミュレーションでは，実に数十万以上のオブジェクトが，同時に動くシミュレータを作っています．（上記の「ゆるシミュ」は，週末研究員としての，個人的な趣味です．）

今のパソコンであれば，数十万から百万オーダのオブジェクトを，メインメモリ上に持つことは，それほど難しくはありません．例えば 100Byte（属性 20〜30 個くらい）のオブジェクトを 100 万個作るためには，単純計算で 100MByte のメインメモリがあれば足ります．（今，ノートパソコンでさえ 4GByte くらいのメモリを搭載しています．）

問題は，オブジェクト間の相互作用を計算するための，組合せ爆発と頻度（回数）です．例えば，1,000 万人都市において，鉄道事故（人身事故）が発生した場合，その事故発生時に鉄道に乗車していた，数万人の人間を，数百パターンもある別の交通手段に誘導するシミュレーションを考えてみます．

たった一つのケースを試すだけでも数千万回（数万人×数百交通手段）のパターンの全てを調べなければなりませんが，当然一つのケースを試すだけでは足りず，これを数千から数万ケース調べることが必要となります．（もう計算が面倒なのでやめますが，）とにかく膨大な計算時間が必要となるのです．シミュレーションを生業としている人であれば，御存じかと思いますが，「1 時間分のシミュレーションするために，3 時間の時間が必要となる」というような話は，結構，普通にあります．

分析や解析が目的であれば，別段問題にならないのですが，例えば，このようなシミュレーションを，リアルタイムで動き続ける社会システムに組み込もうとすると，これは大問題です．

鉄道で人身事故が発生したときに，その 1 時間後の未来の状況（今から 1 時間後に，どの駅で，あるいはどの列車の中で，どれだけの人が，どのくらいの時間閉じ込められているか）などを予測できれば，その 1 時間後の未来に向けて，今すぐに，いろいろ手が打てます．（例えば，臨時バスを事故発生直後に準備し始めることができる，とか．）

これからの社会システムに，トラブル発生後の，訳の分からない近未来（例 :1 時間後の未来）を，数秒から数分で明らかにすることができる「予測」するシミュレーションがあれば，有用となるはずです．

社会システムのシミュレーションでは，常に同じ数のオブジェクトが必要となるわけではありません．例えば 1,000 万人の都市であったとしても，常に 1,000 万人が電車に乗っているわけではありません．ある時刻に限定すれば，同時に電車に乗っている人数は 10 万人を超えることはありません．

私の「ゆるシミュ」では，パソコンの中に作る‘ひと’や‘もの’のオブジェクトは，必要な時間に生成し，役割が終ったらとっとと抹消する，という方式を使っています．これを実現する，超古典的な手法が「動的メモリ」と「リスト構造」です．

「動的メモリ」とは，プログラムの動作中にプログラムの要求に応じて，オブジェクト用のメモリを追加／削除することができる「使い捨てメモリ」です．また，「リスト構造」とは，この「使い捨てメモリ」を連結する「鎖」です．この二つは，リアルタイム性と省メモリが要求される OS 自身や制御アプリでは今でも普通に使

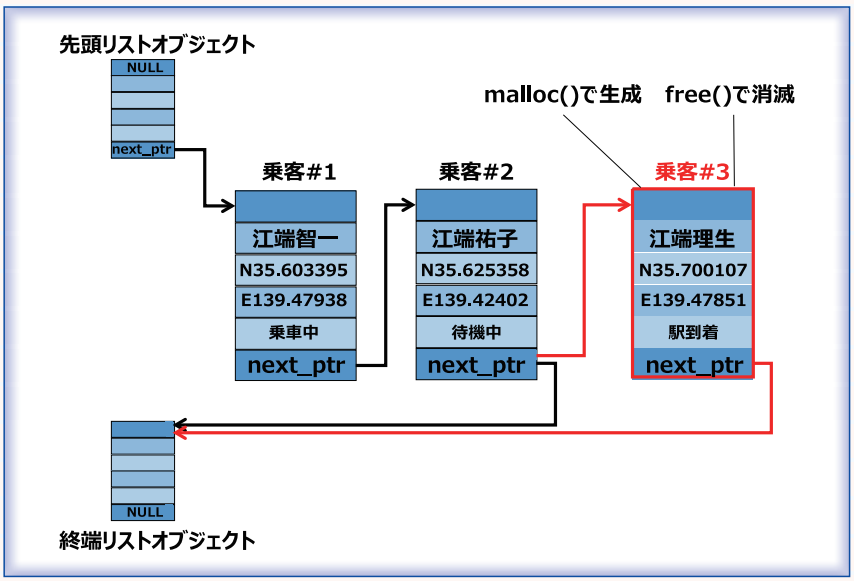

図 15　「動的メモリ」と「リスト構造」の概念

われています（図 15）.

C/C++ 言語では，malloc（），free（）というコマンドで，動的メモリを生成／消滅させ，シミュレーションで必要となるオブジェクトを常に必要最小数にすることができます.

これらのコマンドを使うことで，シミュレーションの驚異的な高速化や軽量化が可能となりますが，使い方を誤ると，OS 領域のメモリすら食い尽くす「メモリリーク」を引き起こします．メモリリークは，軽々とシステムを破壊しますので，古今東西のプログラマから，忌み嫌われ続けています.

しかし，大量のオブジェクトを使った，力ずくの社会システムのシミュレーションを行う場合には，非常に有効で強力なコマンドですので，リスクを覚悟した上で，試してみるのもよいかと思います.

7　社会システムシミュレーションの意義

私は，「社会を理解する方法」には，少なくとも，三つがあると考えております（表 5）．これらのアプローチ（仮説，論理，シミュレーション）には，それぞれメリットとデメリットがあります（表 6）.

表 6 に示しているように，シミュレーション（力ずく）は，少なくとも，私のような

・仮説や論理を打ち立てるほど，頭も良くなく，数学的センスもないが，

・中途半端で未完成の‘もの’（プロトタイプ）を作らせたら，めっぽう早くて，コピペ（コピーアンドペースト）でプログラムを量産できる

—技術者にとって，「社会を理解する方法」としては，なかなか便利なアプローチなのです.

もっとも「シミュレーション」というのは，現実世界との一致を一切保証しない，作り手の自分勝手な世界観に基づく，“自分だけのワンダーランド”ですので，その無矛盾性や完全性について，「論理」には遠く及びません.

それでも，不毛な「仮説」の議論の応酬よりは，はるかに建設的であるとは思います．なぜなら，シミュレーションを本気で動かすためには，そのシミュレーションの中に，自分に都合の良い事実だけでなく，自分にとって都合の悪い事実や，不愉快な事象も組み込まないと，そのシミュレーションは絶対に動き出さないからです.

このように，シミュレーションは，時間も手間も掛かる，相当に面倒臭い方法ではありますが，基本的には，その対象が‘もの’（有体物）でさえあれば，どんなも

表 5　社会を理解する三つの方法

アプローチ	方法
仮説(のみ)	多くの人にとって、納得できる「理由付け」を行う
論理(定式化)	上記の仮説等に基づき、「定式化」する
シミュレーション（力ずく）	仮説も論理もさっぱり不明→パソコンの中に「仮想世界」を作る

表 6　三つの方法のメリットとデメリット

アプローチ	メリット	デメリット
仮説(のみ)	それなりに筋が通っていれば、それなりに正当性を主張できる	「立証の証拠を出せ」「反証の証拠を出せ」という不毛な論争に陥る
論理（定式化）	「実験→観測→定式化→一般化」という、科学的アプローチの金字塔	定式化が無茶苦茶に難しい。高度な数理の知識とセンスがないと、到底届かない
シミュレーション（力ずく）	結論を気にせず、モデル化したオブジェクトをコンピュータの中に作るだけ	モデル化や、パラメータの初期値によっては、現実とかけ離れた結果が、平気で出てくる

のでも試すことができる，ということは大きな利点であると思っています．

8 まとめ

(1) このたび，私のブログや連載を御愛読頂いている，電子情報通信学会の方から寄稿の御依頼を頂き，私が，私なりに社会を理解する方法として―かなり“いい加減”で“アバウト”で“緩い”シミュレーション，「ゆるシミュ」（筆者が命名）」―について御紹介しました．

(2) 具体例としては，「米国大統領選挙のシミュレーション」「残業発生シミュレーション」「江端専用ダイエットシミュレーション」について，その概要を御紹介しました．

(3) たかだか 20～100 行程度の「ゆるシミュ」プログラムが，(a) 現象の大雑把な理解をするのには十分に役に立つこと，(b) 社会に対して 1mm も貢献しなくとも，自分を満足させるのには十分意であること，(c) その問題の対象を理解するには，100 冊の本を読むよりも，はるかに深いレベルで理解することができること，を具体例で説明しました．

(4) 社会システムシミュレーションは，常にスケーラビリティの問題を抱えており，それゆえ，現実世界での近未来予測の道具としては使いにくいこと認めた上で，その問題を解決する手法として，OS や制御アプリの分野で，現役で使われている「動的メモリ」と「リスト構造」の概要を説明しました．

(5) 社会システムシミュレーションの意義は，仮説に基づく不毛な議論を回避することが可能となり，更に，論理化や定式化ができるだけの知性がない私のような技術者であっても，「社会を理解する手段」として十分に役に立つことを説明しました．

謝辞　私の脱稿原稿を校了まで面倒を見て頂き，美しい Web 画面を作成して頂いている EE Times Japan 編集部の担当の村尾麻悠子様，執筆テーマ方針を御教示下さる編集長の竹本達哉様に御礼申し上げます．

江端智一

日本の大手総合電機メーカの主任研究員．1991 入社後「サンマとサバ」を 2 種類のセンサだけで判別する電子レンジの発明を皮切りに，自動車エンジン，映像配信，屋内 GPS，運行管理まで幅広い分野の研究開発に携わる．1993 に “kobore.net” を立ち上げ，週末の趣味の研究やコラムの公開を開始．2012 に Web メディア業界の方から連載の御依頼を頂き，現在に至る．