

「ダイナミックコラボレーション」を科学する

“ Dynamic Collaboration ” from Scientists 'Eyes

江川 尚志* 小林 正好* 山西 健司**
Takashi Egawa Masayoshi Kobayashi Kenji Yamanishi
阿留多伎明良* 並木 淳治***
Akira Arutaki Junji Namiki

要 旨

SCM (Supply Chain Management) を題材に NEC のビジョン「ダイナミックコラボレーション」を定式化してみると、関係する会社数が1, 2社である単純な系でない限りは不安定となることが制御理論から分かります。現実には多くの会社が関係しても不安定が起きないのは、引き込みなどにより自己組織化し、構造を単純化するDynamicsが働いているからであり、このDynamicsを理解することが今後のシステム構築では重要です。また、このシステムが大規模化するときにはシステム内遅延をユビキタスにより抑制することも安定性向上に重要です。これがダイナミックコラボレーションとユビキタスが今後重要になる理由です。

If we formulate “Dynamic Collaboration”, NEC’s business vision, considering Supply Chain Managing (SCM) as an example, control theory shows systems are unstable if the number of business partners is more than a few. This is different from our empirical law. In real world dynamics to form structure exists, which makes systems enough simple to be stable. It is therefore important for us to understand this dynamics in future system design. Ubiquitous computing that suppresses internal system delay is also important for system stability. We believe these clearly show that our future lies in ubiquitous computing and, in Dynamic Collaboration.

1. はじめに

ご存知のように NEC は、2003年ジュネーブでの TELECOM 2003 を始め、機会をとらえ Computer と Networking の融合による、新たなビジネス創造を提案するとともに、ASP、コンテンツ事業者、そして広範なユーザを巻き込ん

だコラボレーションビジネスを提案しています。その象徴が「ダイナミックコラボレーション」です。これこそが NEC の成長戦略の中核を成す事業コンセプトであるからです。

では、「ダイナミックコラボレーション」とは何でしょうか？ 多くの方が IBM の On Demand Business ならご存じでしょう。TV コマーシャルでおなじみのこのコンセプトは、一言でいうならば「事業の需給状況の変化、時代の変化、お客様の嗜好の変化に遅滞なく適応的に、そのサービスリソースあるいはサービス自身を提供すること」です。これにより、お客様としての企業ユーザ様を無駄な設備投資、過大な保守費用、突発的需給バランス不安、などから解放し、本来の業務へ専念できる環境を提供するというものです。

このような動きはコンピュータサービスからすれば当然の成り行きです。しからば、NEC がこのようなサービスコンセプトを提唱するとしたら、どうなるのでしょうか？ NEC はかつてのコーポレート ID, “C&C” から分かるように Computer と Network の2つの領域で多くの製品と、ご満足いただいていた多くのお客様、そしてその運用実績を備えた、世界でも例を見ない会社です。この強みを生かし、サーバ、ストレージなど基本要素をネットワークで広範に関連付けることにより、単なるリソース共用ではなくサービス・パートナーとの連携すら動的に最適化させることができると考えています。このコンセプトが「ダイナミックコラボレーション」なのです。

ネットワークについていえば、単なる企業ネットワークに限らず、国内No.1, 3Gの世界では世界有数のシェアを誇るモバイル領域から、無線LAN, 広域イーサネット, WAN, 公衆網, 躍進著しい ADSL, ITS などあらゆるアクセスを念頭においたものであることはいまでもありません。

2. 「ダイナミックコラボレーション」をどう記述する？

図1の左側をご覧ください。A社はB, C, D社と連携して

* システムプラットフォーム研究所
System Platforms Research Laboratories
** インターネットシステム研究所
Internet Systems Research Laboratories

*** 支配人
Vice President

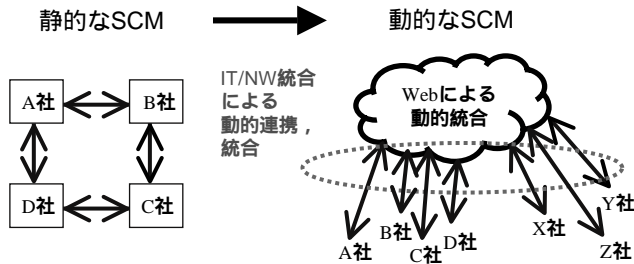


図1 SCMにダイナミックコラボレーションを適用する
Fig.1 Dynamic Collaboration in Supply Chain Management.

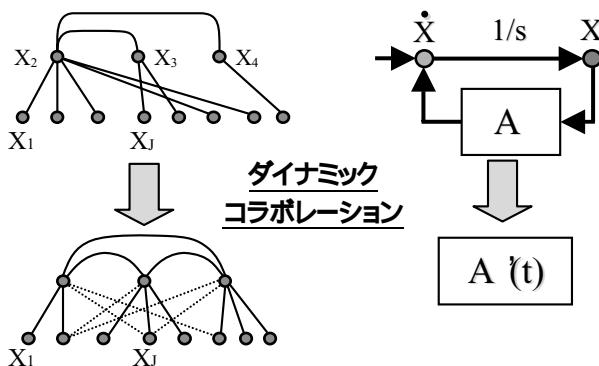
あるサービスを提供しているとしましよう。いわゆる Supply Chain Management (SCM) です。この4社の関係は一朝一夕にできたわけではないので、一般的には十分長い期間を見て静的なものでした。ところが、昨今のOPEN化、世界的水平分業化が進むと、A社から見てB, C, Dに求める機能がグローバルに選択可能な状況が生まれつつあります。たとえばB, C, D社の代わりにX, Y, Z社を使うことが可能になってきました。

図1の右側はまさにそのような状況を表しています。A社は、自己の提供するサービスの要素機能を、広く全世界から電子の速さで調達し、その時点でコスト的にも、性能的にも最適なサービスを提供し、自ら享受しています。これは、図1左を静的なSCMと定義すれば、右側は動的なSCMであるといえます。これを1つのダイナミックコラボレーションのモデルとして科学していきます。

3. ダイナミックコラボレーションを定式化する

今、N社の連携 (N個の要素からなる Supply Chain) を考えてみましょう。

図2左上をご覧ください。連携するN社を表すN個の「○



3次以上の系は一般には大域的安定とならない。ダイナミックコラボレーションによりAが時間的に変化するとき、系の安定性は保証できるのか？

図2 SCMに適用したダイナミックコラボレーションを制御理論から見る

Fig.2 Control theory's view of Dynamic Collaboration in Supply Chain Management.

(以下ノードと記述する)」が書かれており、各社の関係はその間に張られた結線で表されます。各結線は双方向の場合もあるし、単一方向の場合もあります。厳密に議論を進める上では、各結線を単一方向で定義したいのですが、いたずらに図面が複雑となるので、双方向性が定義されているとしましょう。すなわちノード X_i はノード X_j とは別個にノード X_2 、ノード X_3 に收容されているので、独立のようですが、実際にはノード X_2 、ノード X_3 間の結線を介して相互に影響を及ぼし合っています。

図2左下は、ダイナミックコラボレーションによって、各社の連携が変化したことを表しています。注意する点は、N個のノードの間の結線のつながりかたは確かに瞬時に変わっていますが、ノード自身の表す状態 (たとえば資本金、剰余金、棚卸、負債など) はそのまま引き継がれるという点です。これは至極当たり前のことで、連携を変えたところでそれだけで一瞬にして各社の資産が雨散霧消してしまうはずはないからです。それらは、新たな結線がもたらす入出力の結果、じわじわと変化していきます。そこでこれらXを近代制御理論では「状態変数」と呼んでいます。結線と状態変数は、回路の世界でいえば、記憶機能のない抵抗Rと、電荷や磁場などの記憶機能のあるC, Lが各々対応します。

このように、N社の連携をN個の状態変数と、各状態変数間の連結で表せるとすれば、それらのシステムのDynamicな動向は図2右上のように $\dot{X}' = AX$ なる微分方程式の時間解として与えられます。ここでAはN個の状態変数間の結線状態だけで決まる $N \times N$ の行列です。すなわちAが与えられれば、N社の連携状態が分かるリストのようなものと考えてください。

さて、図2左下で説明したように、ダイナミックコラボレーションによって、連携N社の接続がバタバタと変更していくわけであるので、これを図2右上の表記でいえば、まさに行列Aが時変的に変化することができそうです。すなわち $\dot{X}' = A(t)X$ なる時変・微分方程式で表せるということになります。

4. 試行的ダイナミックコラボレーションはシステム不安定を誘発する？

いかなる複雑なシステムでも、いったん図2のように定式化されてしまえば、それらのシステムの挙動を把握することは比較的容易です。ここでは、システムの最も本質的の検証として、その安定性を検証します。

まず、おとなしいシステムから考えてみましょう。いいかえれば、それらの構成要素はすべて「3つ殴られても黙って我慢するか、せいぜいその半分程度の反撃しかしない連中の集まり」と考えればよいでしょう。図3右上をご覧ください。曲線 α はこのようなシステムのある種のポテンシャル曲線を示しており、通常ポテンシャルの最も低い凹点に停留しています。簡単には止まった時計の振り子を想像してもらえばよいでしょう。ここで、システムの結線が突然

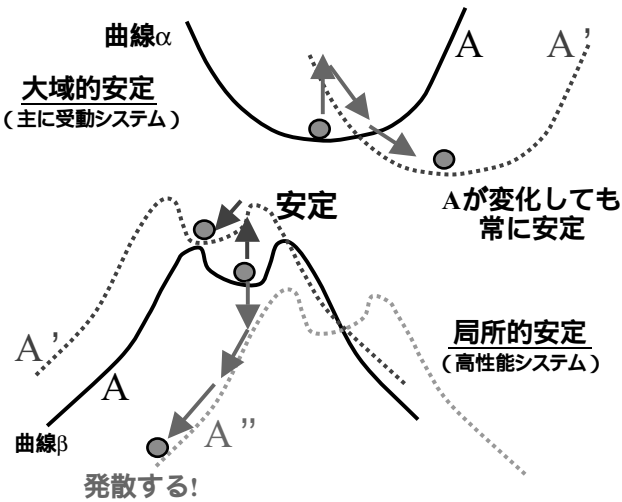


図3 大域的安定，局所的安定となるポテンシャルの形
Fig.3 Potential curve on global stability and local stability.

変化し、それに従い行列 A が A' と変化したとしましょう。一般的には N 個の状態変数間の力のバランスが変化したのですから、今までの停留点はたちまち傾斜点と化してしまうわけです。乱暴な比喻ですが、止まっている振り子時計を箱ごと急に動かすことを考えてください。システム自身がおとなしい場合には、その変化がどんなに大きくとも、必ず新たな停留点に向かって、安定的に収束していきます。このような絶対的安定を制御の世界では大域的安定と呼んでいます。

さて、今度は現代のコンピュータ社会を支えている高性能システムを考えてみます。

図4をご覧ください。図4はおよそ世間でいわれている高性能システムの共通の特性を示したものです。それらはたとえば、変調方式、コーディング方式、他変数制御などを連想してください。図4横軸は、そのシステムの動作環境の良しあしを、縦軸はシステム特性の良しあしを示しています。図中45度の直線はおよそ何の工夫もない自然システムを示しており、動作環境の良しあしに応じて特性も安定的に連動するというものです。

一方、高性能システムとは、おなじ動作環境においても、先の自然システムより数段良好な特性を提供するものです。ただし、その特性改善は一定の動作環境を前提に様々な創意工夫が成されたもので、その前提条件が崩れるとシステムの破綻をきたし、自然システムより悪い結果しか生まない状況に陥ります。これを一般的には「しきい値効果」と呼んでいますが、このしきい値は、高性能システムほど動作環境が良好な方向へシフトしていきます。これが超高性能システムの動作環境はピンポイントといわれる理由です。

ちなみに、動作環境を右の方向にドンドン改善していった場合の振る舞いを検討してみると、自然システムは、たとえばアナログレコードのように、どこまででも良い特性が発揮されます。一方、昨今のデジタル処理の場合には有

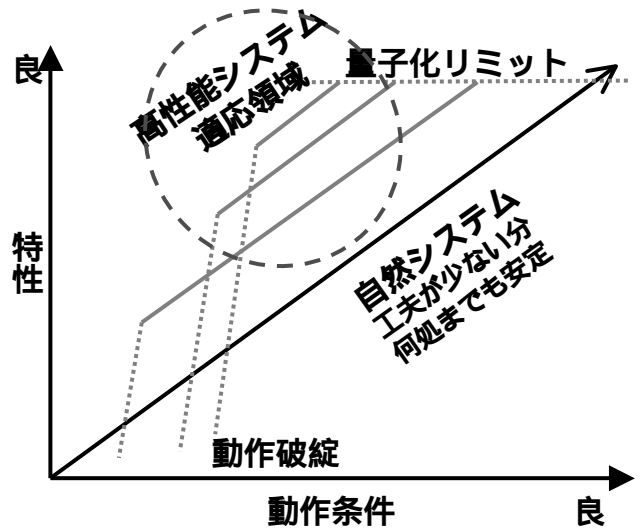


図4 高性能システムの一般的特性とは？
(変調方式、コーディング、多変数制御等々)
Fig.4 What is a high-performance system, in general?
(modulation, coding, multi-parameter control, ...)

限の処理能力をいかに有効に活用するかとの観点から、一般的には特性リミットが起きることを容認しています。

以上のことから、高性能システムの適性動作環境は、大域的でなく、局所的なものであることが分かります。

図3左下に戻ってください。今ダイナミックコラボレーションの結果、行列 A のポテンシャル曲線 β が、瞬時に A' に変化したとしましょう。ポテンシャル曲線 β は局所的安定ですが、この場合には運良く安定領域へ再び戻ることができました。一方 A からさらに大きく変化して A'' の場合を見てみると、この場合には不安定な左下がり傾斜からの再スタートとなり、発散の憂き目を見ることになります。

より一般的な場合についていえば、 $X' = AX$ のシステムが安定かどうかの検証は、厄介だが十分条件というのであれば、 A 行列がある種の特性を持っていれば保証されます。しかし、ここでいうダイナミックコラボレーションにおいて行われるような、試行的接続変更が本当に行われたとすれば、連携する会社の数 N が2~3程度のコンパクトのシステム以外、間違いなく不安定現象を起こす。…はずであるのに、余りこのことが騒がれません。なぜなのでしょう？

いよいよ、ダイナミックコラボレーションの本質に迫ります。

5. ダイナミックコラボレーションの本質に迫る

今まで、一般的多変数システムの安定性は極めて限定的であることを述べました。そのなかでさらに、システムの構成をDynamicに変更させるというようなことが可能なのでしょうか？

これは極めて難しい問題に見えます。そこで、私達はここで、与えられたシステムが安定かどうかを考えるという手法から、安定に存続している多変数システムの秘密に迫

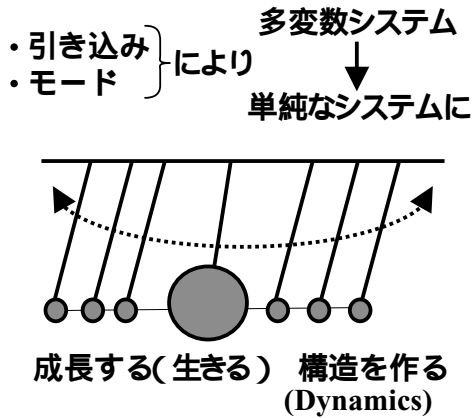


図5 なぜ発散しない?
Fig.5 Why not diverge?

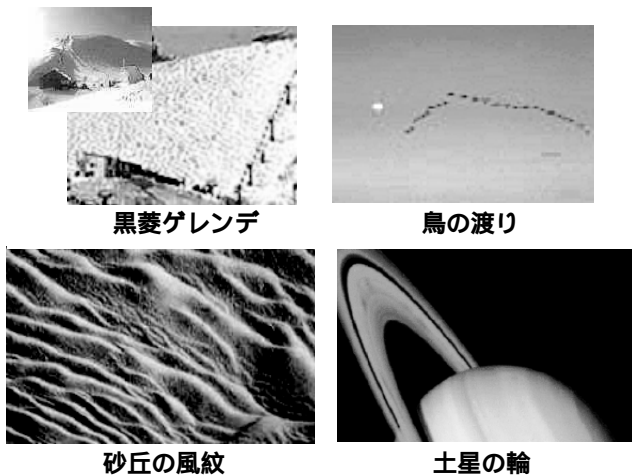


写真 成長は構造を作り, 死ぬと分解される

Photo Growth makes structure. Death brings decomposition.

ります。

図5中央をご覧ください。大きな振り子に小さな振り子がたくさん付いて動いて従属的に振れています。この系は変数は多いが、システムの振る舞いを決定しているのは実は中央の大きな振り子のみの単純な系と考えることができます。自然界や物理現象において、多くの要素が本来取り得る自由度を放棄して、単純な姿態に収束する現象があります。物理の言葉でいうと「引き込み」や「モード」です。すなわち多変数システムが縮退して単純システムとしての振る舞いをしているのです。

多くのものが、安定的、永続的に生き延びるために、おのおの自由をかなぐり捨て、ある種の単純構造を作っているのです。ある種の蛍は夕闇が迫ってくると、1つ2つとばらばらに点滅を始めますが、気が付くと無数の固体が同期して点滅をする状態になるそうです。これは、個々が点滅パラメータの独立性を放棄し点滅周波数と位相を共有することにより、なるべく遠方からでもその存在、この場合は点滅周波数を同種の異性に認識させるためと推察されて

います。だからこそあのか弱い蛍が生き延びているわけですが、この同期化は蛍のDNAにプログラムされており、「構造」を作る鍵となっています。

構造を作るのは何も蛍の点滅に限った話ではありません。写真をご覧ください。左上の写真は冬の八方尾根ゲレンデであり、初心者スキーヤを悩ませる、大きなコブが一面に存在しています。これは決してゲレンデの地形がこのようなコブに成っているのではありません。このゲレンデは毎朝、大きな雪上車でコブをならしています。しかし、日が高く上る頃にはスキーヤ自身のまったく気ままな滑走によって、このような規則的なコブが成長してくるのです。左下の砂丘の風紋もそれに似たところがあります。砂丘は淡々としたなだらかな曲面とそこに吹く風だけがすべてです。図の右上の渡り鳥の編隊飛行の様子をご覧ください。航空ショーでもないのにこのような編隊を組んで飛んでいます。土星の輪がレコード盤のように薄く平たい平面軌道状の小天体であるということも興味をそそります。小惑星どうしの衝突摩擦がこのような形を生みます。

一般に、集合体の規則性は、時間の関数で消滅していくものとエントロピー増大の大法則が教えています。しかし、これは系へのエネルギー注入が隔絶している場合に限るとこのようになります。一般的な自然景観あるいは生物環境においては、太陽エネルギーがもたらす風、波、雨、温度変化、地球の核からの火山活動による噴火、温泉など様々なエネルギーの出入りが存在します。このような開放系では、実は、構造を作ることこそが、エネルギー的にも安定で、永続を約束する姿態になる、と散逸系や協同現象の解析はわれわれに教えます^{1,2)}。いいかえれば、普通に存在する低エネルギー状態で安定的に存在が保証される姿態には、特有な構造が対応するということです。様々な姿態があるとしても、上記姿態を取った後は、それを離れるためには今以上のエネルギーを必要とするのであるから、そこが系の停留点となるのです。

ビジネスの世界にいいかえれば、あるシステムを運用、管理するのに、それを維持する費用を系へのエネルギーとして、管理経費を考えれば分りやすくなります。同じ使うなら、安価でシンプルな物が生き残るということです。多少、性能が良くても、高価で使いにくい物は商品として残らないということになります。

いいかえれば、ダイナミックコラボレーションにより、ますますビジネスが繁栄し、永続的發展をすれば、それは雑多な個の集合ではなく、必ず構造を持った集合体に帰着するということが予想されます。この構造が破壊するのは、その集合の寿命が尽きて死を迎え、エネルギーの出入りがなくなった時といえましょう。

次に、構造のキーになる物は何であろうかという点を検討してみましょう。

6. ネットワーク・アーキテクチャに見るその構造変化

第5章で述べた「構造を持った集合体」の例としてネットワークを取り上げて、その構造変化のキーになるパラメータとそれらの関係、すなわち「Dynamics」にメスを入れてみます。もちろん、前述した蛭や八方尾根ゲレンデの例のように、構造を作る力学が限られたパラメータで記述できるわけではありませんが、それぞれの構造の本質を抽出するためになるべく少量の変数に着目して考察を与えます。

6.1 人為計画型(1): 階層型(従来の公衆電話網)

図6の左図は、公衆電話網に代表される人為計画型のネットワークです。網の形状から階層型ネットワークとも呼ばれています。このネットワークの特徴は、その計画的構築手法にあります。たとえば、全国網を構築するには5ヵ年計画、10ヵ年計画、あるいはその国家の事情によってはもっと長期間にわたる設備投資計画を立案します。そして、年度ごとに計画的に交換機や加入者線・伝送路を整備し、サービスエリアの拡大をしていきます。この計画方法がこの網における“Dynamics”です。典型的な計画方法は、待ち行列理論すなわちRandom Walkに代表される確率論的取り扱いに基づいた構築計画です。その結果として、たとえば、とある加入者線交換機に入力される加入者からのトラフィック(この場合は、「呼」と呼びます)はポアソン分布に従って発生し、さらには、その宛先もランダムであり、とある確立分布に従って上位の中継交換機に到達する、などなど。典型的には発生呼の平均12.5%が市外番号を指定したトラフィックである、など。この結果として、ピラミッド型の階層型ネットワークが計画されました。

番号計画もすべて事前計画法が基本にあります。誤解を恐れずに単純に表現してしまえば、たとえば日本では、桁数はそれぞれ異なるものの市外局番は地域ごとに階層化されているのはご承知のとおりで、宛先電話番号の頭の部分を見ればどの地域へのトラフィックであるのか瞬時に判断できます。アメリカではどんな大都会でも過疎地域でも市外局番は3桁と決まっており、初めの3桁だけでどの交換局に繋がればよいのか判断できます。

この事前計画法に基づくDynamicsは、スクラッチでそれまでになかったネットワークをまったく新規に構築する

には、実に効率的でした。その代わり、新興団地が急に開発されたり、衛星都市などの市外局エリア間トラフィックが見込みより多かたりすると、新規加入を待たされたり、話中ばかりでなかなか繋がらなかったり、など加入者の不満を増加させてしまった経緯があります。このため、事前計画も年度ごとにピラミッド形態を部分的に崩す斜め回線を追加したりと、修正しながら実施されました。

6.2 人為計画型(2): メッシュ=ランダム型(初期のコンピュータ網)

80年代後半に出現したジオデシックネットワーク議論は、上述したネットワーク構築論に一石を投じました。ジオデシックとは「測地線」のことであり、2地点間を最も短距離で接続することを旨としたDynamicsです。結果として、メッシュ型に接続された多数のノード、それも仲良しどうしが直接接続された網の形態をとることとなりました。図6の右図に示したように、典型的には同質なノードが必要に応じたリンクでメッシュ状に接続されています。

90年代に入ると、地域ネットワークの構成方法と、さらには地域キャリアの独占性議論にまで発展することとなりました。しかし、このDynamicsが最も直接的に影響を与えたのは初期のインターネットだったのでないかと考えます。Metcalfeなどが70年代に考案したイーサネットは、たとえば同軸線に複数のタップを立てて接続するブロードキャストを基本としていたのですが、これはフルメッシュの最も典型的な構築方法でした。しかし、UTPを用いて端末を個別にEther-HUBに接続する10BT方式がLANに導入された90年代は、この個別線方式を用いてどのように仮想的にフルメッシュ網を構築するかの議論が盛んでした。

また、それまでの研究用にのみ許可されていたインターネットが商用に開放されて、その結果として爆発的にユーザ(加入者)が増加したのも90年代初頭のことでした。すなわち、すでに確立されていた電話網を活用して、モデム呼として音声と透過な帯域を確保しつつも、地域や電話局とはまったく独立な論理的アドレス体系(IPアドレス)を確立してジオデシックなルーティングによるサービスが始まったのです。この、電話網という物理的接続と、IPルーティングという論理的接続の根本的Dynamicsの違いから、「1996年末にインターネットは崩壊する」と1995年に予言したのは前述したインターネットの大家Metcalfeでした。しかし、現実にはインターネットは崩壊せず、ますます盛隆を極め、その結果、「もし予言がはずれたら掲載論文を食する」と明言した本人が公衆の面前でそれを実行する羽目になったのは、米国人独特の演出だったのではないかと、いう説もあります。

6.3 成長型: Scale-Free 網(現インターネット)

なにゆえにMetcalfeの予言がはずれてインターネットが崩壊しなかったのか、については後述するとして、まずは、図7をご覧ください。Scale-Freeと呼ばれるDynamicsで

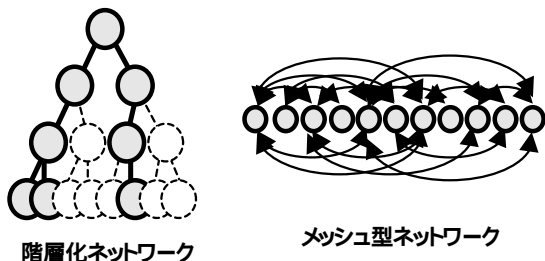
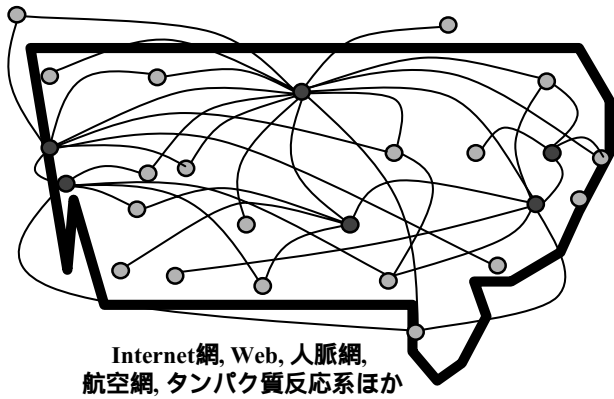


図6 従来の計画的なネットワーク
Fig.6 Traditional preparatory networks.



Internet網, Web, 人脈網,
航空網, タンパク質反応系ほか

図7 スケールフリー網
(自然界で多く見られる成長型網)

Fig.7 Scale-free networks (growing networks common in natural world).

1999年に提案されたものです³⁾。自然界に多く存在し、WWWの論理網構造や人体構造、はては宇宙構造から生命をつかさどる化学反応、人脈確立の方程式までを合理的に説明しようとするものです。基本的考え方は、「初めに形があるのではなく、とあるDynamicsに従って、それぞれのノードが自己進化していく」となります。

ノードを一匹の蛍と仮定して、前述した同期点滅蛍光の例で説明しましょう。

階層型の人為計画式ならば、誰か中心に強烈なリーダーシップを持つ蛍がおり、何らかのメカニズム、たとえば自分の点滅周波数に合わない他の蛍は食べてしまうとか、に従って計画的にかつ強制的に発光同期を取ります。メッシュ=ランダム型の人為計画式ならば、朝までかかっても点滅が同期することはなく、遠くから見れば平均的にはボツと光る薄暗い蛍光が異性の蛍に見えるだけです。これが成長型のScale-Freeですと、各蛍のDNAに「発光タイミングは受光タイミングに合わせる」というDynamicsがプログラムされているので、各蛍が同質なものであっても、独立した発光変数を放棄して同じ周波数と位相で発光する、というものです。

ちなみに、コンサートなどでアンコールを求める聴衆の拍手が、初めはバラバラであるのにすぐに同期した拍手になり、いよいよ指揮者が舞台の袖に現れると一段と大きなバラバラな拍手に変化することは、読者の皆様も経験されたことがおありかと想像します。これもスケールフリーで説明すれば、最初は各ノードのエネルギーが低いのでDynamicsに従ってただちに同期したのが、指揮者出現という環境(変数)の変化で各ノードにエネルギーが注入され、自由に行動しようとする力が同期させるDynamicsの力を上回ったため、局所的安定点から少しずれた状態で安定した、ということになります。

では、Metcalfの予言はなぜ外れたのか。図8の左図をご覧ください。初期のインターネット構造であるジオデシ

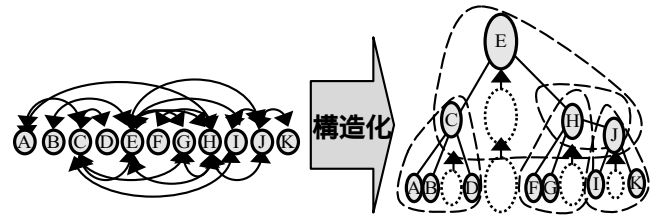


図8 メッシュ網からスケールフリー網へと成長
Fig.8 Growth toward scale-free networks.

ックそのものです。ここで、AからKの各ノードは、たとえばネットワークでいえばルータやスイッチ、WWWでいえばWebサーバといえます。ただし、このノードはルータなどの機械だけではなく、SIerやオペレータ、アドミニストレータなどの人的資源をも含んでいます。ジオデシクのDynamicsだけであれば、この網構造は何があっても変化はしません。したがってどんなにユーザが増えても、今あるサーバはそのまま存在するだけ。しかし現実には何が起こっていたのか? 「最近このWebサーバのCPU使用率が高いな〜」「そろそろもう一台、サーバ立てとくか…」このようなアドミニストレータの判断により、そのWebサーバのレスポンスが向上する。するとエンドユーザは、「あのサイトはサクサク返事が来て気持ちがいい」となって、結果としてまたアクセスが増加する。すると…この循環がそのWebサーバをさらに強力なものとし、結果的にそのサーバは他のサーバを従えたハブとなるのです。図8右図でいえば、E/C/H/Jの各ノードがそれに相当します。このDynamicsはサーバだけではなくルータのアドミニストレータにも同様にいえることです。その結果として、IPアドレスの学習数が多く帯域の大きなルータサイトがハブとしてでき上がってくるのです。

図8右図は一見すると、まるで階層化されたネットワークに見えるかもしれませんが、それは結果的にそうなっただけであって、前述したDynamicsと直接は関係しません。ことの本质は、ネットワークやシステムのアーキテクチャが先に計画されるのではなく、どんなDynamicsがネットワークやコンピュータシステムを決定付けるか、そのDynamicsにインパクト(エネルギー)を与える環境(市場など)は何なのか、そのDynamicsを実現するシステム実装は何なのか、です。

7. ダイナミックコラボレーションの本質とは

以上、述べてきたようにダイナミックコラボレーションの本質は、何かお仕着せのコンセプトに基づくアーキテクチャ論ではなく、構造を作っていく「Dynamicsの入れ物」を提供する物と理解する必要があります。砂丘にいくら綺麗な文様を書いたところで、翌日には風と砂丘の連携が作る風紋に被われてしまいます。「Dynamicsの入れ物」とはこの場合、砂丘の曲面や、砂粒の種類、大きさ、風をどの

ように吹かせるかなどです。実業の世界に目を転ずるとこれはまさに、市場環境そのものということになります。顧客の嗜好、金融環境、景気、季節、イベント、国際関係、安全・・・などをすべて勘案し、関与するものの出費(エネルギー)が最小になるような系が生き残るのでしょう。

これがダイナミックコラボレーションの本質であり、決してある種の万能アーキテクチャ論ではないことがお分かりいただけたと思います。

8. ダイナミックコラボレーションからユビキタスへの必然

今まで、大規模システムの振る舞いについて、色々検討して来ましたが、システムの安定性を左右する重要な要素、「遅延」の影響について考察しておきます。

図9は、先に示した図2の各結線に遅延(情報伝達や判断、行動の遅れ)がある場合を示しており、各結線には τ_i なる遅延要素が直列に繋がれています。これこそが現実のシステムです。

図10は状態変数が1つの最も単純な系の収束状況を $X-\dot{X}$ の位相面に記したものです。 $\tau=0$ の場合には、一番内側の直線のように一直線で安定点に収束していきませんが、 τ が非零でどんどん大きくなるに従い、いわゆるオーバーシュートをしながら収束するようになります。しまいには収束どころではなくなっていく様子がよく分かります。この遅延の影響は、システムへの制御がどのくらい早く影響を及ぼすかの、いわゆるシステムの時定数 τ_0 との比で利いてきます。ぶらぶらと散歩する場合、障害物に気づくのが多少遅れたとしても何とかありますが、高速道路を時速100kmで走る場合には一瞬の判断の遅れが重大事故に繋がるのと同じです。よって、ダイナミックコラボレーションの効果を向上すべく、迅速な変転システムを作るにしたがって、それに関与する多くの動作にかかわる遅延の影響が顕在化してくるのです。

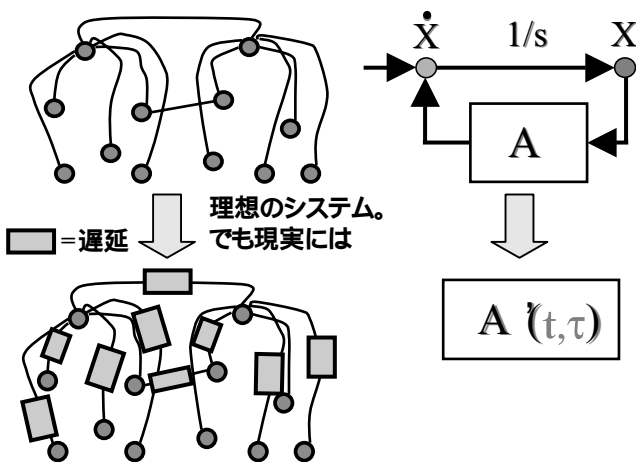


図9 システムの安定性への遅延の影響
Fig.9 Effect of delay against system stability.

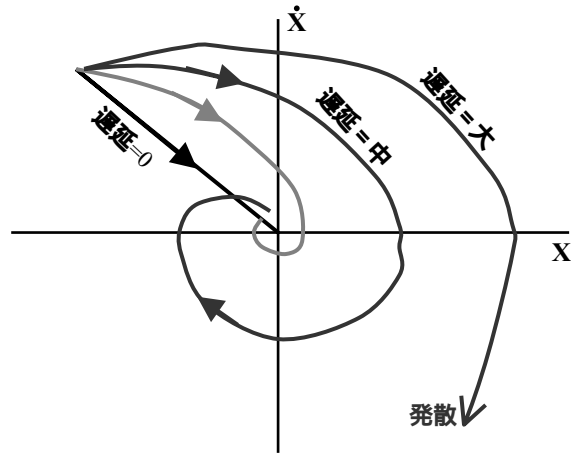


図10 収束への遅延の影響
Fig.10 Effect of delay on convergence.

9. なぜ、遅延が怖いのか?

図11をご覧ください。左上はいわゆる「やまびこ」を示しています。その下のTV信号のゴーストも同じです。遅延は、多くの過去の虚像を発生させ、それが見かけ上システムを複雑に見せるところが問題なのです。同図右側の交通事故の例は多少、意味合いが異なりますが、最もインパクトの強い問題なので挙げさせてもらいました。すなわち、交通事故原因の約50%は危険状況の発見遅れで、それが過剰反応を呼び操作ミスを起こさせるということです。

このように書くと、「いや、遅延ではなく、情報の齟齬、すなわちデータ化けや紛失も問題なのだ」とお思いになる読者の方もいらっしゃるかと思います。しかし、現実のシステムにおいては、特にインターネットにおいてはエンドエンドのTCPセッションがデータのインテグリティを確保しており、何度も再送をかけるなどの通常の処置を施せばデータ化けによる情報状態の不一致は、時間をかければ回避

図11 遅延はシステムを複雑に見せ、過剰反応を呼び(そして致命的な操作ミスを引き起こす)
Fig.11 Delay makes a system appear more complicated than it really is (and often leads to fatal erroneous operations).

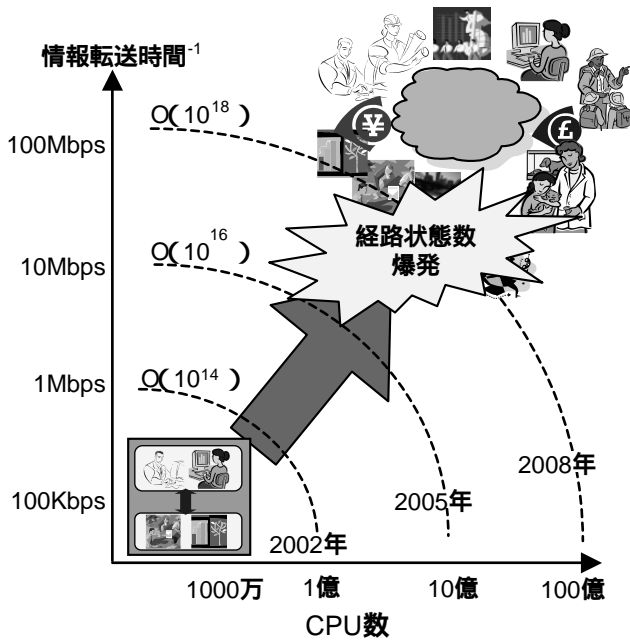


図12 ユビキタス環境下での状態数の爆発

Fig.12 Explosive increase in number of states in ubiquitous world.

できるのです。むしろ、この「時間をかける」ことにより、前述した「遅延による状態の不一致」を誘発し、これがシステムを不安定化させる主要原因になる、と考えるわけです。

実際、第10章のユビキタスの世界になると、図12に示すように、たとえば家電などに実装されたCPU間のネットワーク接続数は現在の1万倍以上に爆発すると予想されています。このような大規模システムにおいては、どのようにしてコストをかけずに遅延を最小にするかが技術的課題となるのです。

10. なぜ今ユビキタスなのか？

大システムの安定性に関与する2大要素は、今まで議論してきたように、1に構成要素数、2に遅延です。一般的に遅延が小さく、構成要素が少ない領域は安定です。ダイナミックコラボレーションが高性能になればなるほど、参加者の操作遅延が問題になり、システムを不安定にさせます。それを救うのが「いつでも、どこでも、何とでも」の「ユビキタス」技術です。これこそが、将来の本格的ダイナミックコラボレーションの遅延不安定を救う不可欠な環境なのです。

前述したMetcalfeの予言を救ったのはSIerやアドミニストレータのノウハウでした。しかし、ユビキタス世界になり、接続経路数が1万倍になったからといって、アドミニストレータやSIerの数が1万倍に増加するというは現実的ではありません。事実、そろそろSIer不足を懸念する声が出始めています。ユビキタス技術は、この状態数爆発による変数急増と相互情報連携の遅延を解決する手段であるといっても過言ではありません。

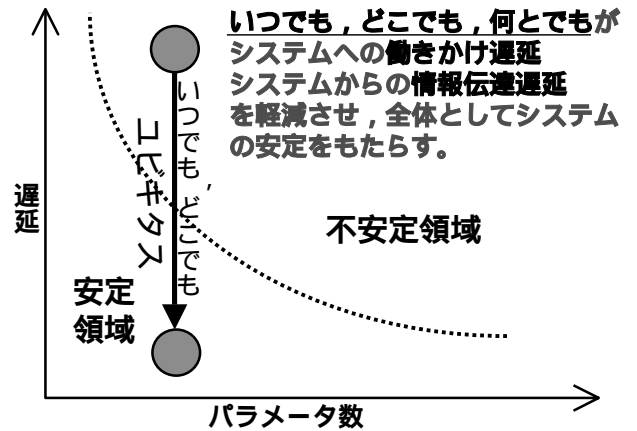


図13 ユビキタスはシステムの内部遅延軽減のため不可欠
Fig.13 "Ubiquitous" is indispensable in suppressing internal system delay.

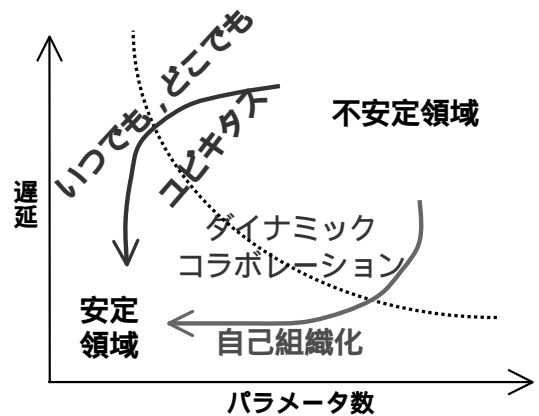


図14 したがって、広域システムには両方が不可欠!
Fig.14 Wide-area Dynamic Collaboration requires both!

図13はまさにそのことを示しており、図14は初めに述べた「安定なシステムは構造を作る」という、いわゆる自己組織化がシステムの複雑さからくる不安定要因を軽減している様子を示しています。例を挙げてみましょう。ある系に重要な判断を下すキーパーソンが出張中、あるいは長期間デスクを空けていたとしましょう。すると、重大な環境変化が起きても、的確な判断ができずシステムパラメータの最適化がなされないこととなります。するとその系に関係する多くのシステムは大きな遅延を抱え込んでしまうこととなります。さらに、昨今の金融システムの場合、そのアクションは秒単位です。このような、極めて短い時定数のシステムを内包するようになれば、まさに系の安定性は系内遅延が握ることとなります。将来の広域的ダイナミックコラボレーションが真に社会の中核を担うとすれば、この

- ・自己組織化
 - ・系内遅延
- を忘れることはできません。

11. おわりに

冒頭に、IBMのOn Demand Businessの狙いとして、『事業の需給状況の変化，時代の変化，お客様の嗜好の変化に遅滞なく適応的に，そのサービスリソースあるいはサービス自身を提供すること』です。これによりお客様としての企業ユーザ様を無駄な設備投資，過大な保守費用，突発的需給バランス不安などから開放し，本来の業務へ専念出来る環境を提供」と書きました。

この文章とダイナミックコラボレーションの本質として書いた下記の部分

「ダイナミックコラボレーションの本質は，何かお仕着せのコンセプトに基づくアーキテクチャ論ではなく，構造を作っていく『Dynamicsの入れ物』を提供するものと理解する必要があります。砂丘にいくら綺麗な文様を書いたところで，翌日には風と砂丘の連携が作る風紋に被われてしまいます。『Dynamicsの入れ物』とはこの場合，砂丘の曲面や，砂粒の種類，大きさ，風をどのように吹かせるかなどです。実業の世界に目を転ずるとこれはまさに，市場環境そのものということにあります。顧客の嗜好，金融環境，景気，季節，イベント，国際関係，安全・・・などなどをすべて勘案し，関与する物の出費（エネルギー）が最小になるような系が生き残るのでしょう。

これがダイナミックコラボレーションの本質であり，決してある種の万能アーキテクチャ論ではありません。」

これを比較して読んでいただければ，ダイナミックコラボレーションがより広範な観点でOn Demand Businessを包含していることが分かります。さらに，今後それらのシステムが全国的，全世界的な広がりの中で，金融指数のような極めて短時間で膨大な財を運用するシステムまでをも含めた動的システムを提供するようになった時には，系の遅延問題こそが最大のネックとなります。NECはこの問題を，ユビキタス技術で克服できる世界で唯一の会社であることを明記して，この論文を閉じたいと思います。

参考文献

- 1) ハーマン・ハーケン，「協同現象の数理：物理，生物，化学的系における自律形成」東海大学出版会，1980年4月。
- 2) 清水博，「生命を捉えなおす 生きている状態とは何か」，中央公論社，1990年10月。
- 3) アルバート＝ラズロ・バラバシ，「新ネットワーク思考 世界のしくみを読み解く」，日本放送出版協会，2002年12月。

筆者紹介



Takashi Egawa
えがわ たかし
江川 尚志 1991年，NEC入社。現在，システムプラットフォーム研究所主任。電子情報通信学会，IEEE各会員。



Masayoshi Kobayashi
こばやし まさよし
小林 正好 1997年，NEC入社。現在，システムプラットフォーム研究所主任。電子情報通信学会，IEEE各会員。



Kenji Yamanishi
やまにし けんじ
山西 健司 1987年，NEC入社。現在，NECインターネットシステム研究所主席研究員。電子情報通信学会，人工知能学会，情報理論とその応用学会，ACM，IEEE各会員。



Akira Arutaki
あるとき あきら
阿留多伎明良 1980年，NEC入社。現在，システムプラットフォーム研究所長。電子情報通信学会，IEEE各会員。



Junji Namiki
なみき じゅんじ
並木 淳治 1972年，NEC入社。現在，支配人。電子情報通信学会会員。