

インターネット技術のエッセンス —COVID-19 の経験から

村井 純¹ (慶應義塾大学)、
長 健二郎² (株式会社インターネットイニシアティブ)

要 旨

新型コロナウイルス感染拡大で予想もしなかった環境変化が起こり、あらゆる人と組織があらゆるレベルで対応を余儀なくされた。その中で社会が直面した問題は、激変する環境にも迅速かつ柔軟に対応できるように、社会システムを再構築することである。本稿では、その実現にはまずデジタル社会の価値観を社会全体で共有する必要があるとし、自由と創造を育み変化に強いデジタルの仕組みを、インターネット発展におけるアーキテクチャとインターネット文化の役割から解説し、デジタル社会に向けてデジタルのメリットを如何に社会に組み込んで行くべきかを考察する。With/After コロナ時代におけるデジタル技術のもっとも大切な役割は日本の社会が健全に発展する基盤を提供することである。

キーワード：インターネット、デジタル社会、アーキテクチャ、エンドツーエンド

1. 環境変化としてコロナ禍を捉える

コロナ禍で IT 化の時計が進んだ。一般利用者に Zoom や Netflix が一気に浸透し、オンライン会議やリモートワークがあたりまえになった。その一方で、さまざまなデジタル化の遅れや問題点が露呈した。日本のデジタルはインフラは強いのにサービスが弱いという不満と失望の声が寄せられた。利用者視点では、コロナ関連の行政サービスに関して、給付金や協力金の申請処理や支給を巡る混乱、ワクチン接種の調整における省庁間や国と地方の連携の不備が目立ち、コロナ対応のアプリやサービスも完成度が低かった。行政側から見ると、他国との比較においても、いまだに紙や人手による処理が多く、組織ごとのデータ管理がばらばらで連携できないことなどから非効率で、かつ、何をするにも法的な制約が多い。

デジタル化の問題を考える場合、コロナ禍を大きな環境変化と捉えることが適切である。非常事態として平時と区別して考えるのではなく、今の社会は常に変化の中にある前提で、日頃から変化に適応し、大きな変化にも備える必要がある。新型コロナウイルス感染拡大で、予想もしなかった環境変化が起こり、あらゆる人と組織があらゆるレベルで対応を余儀なくされた。柔軟に適応できた部分とそうでない部分があり、変化に追従出来ない組織やルールが炙り出された。それらの組織や制度の多くは、そもそも変化に対応する前提で作られていない。

問題はデジタルというより社会システムの硬直化である。これまでのやり方では未曾有の事態に対応できないことが明らかになった。日本の社会が直面しているのは、硬直化した

¹ 慶應義塾大学教授

² 株式会社インターネットイニシアティブ 技術研究所 所長

社会システムをいかに活性化し、激変する社会環境に適応するよう再構築するかという課題である。その解決のいとぐちはデジタル改革にある。デジタル革命の本質は、情報の流通によって異なる考えや文化が交流し、柔軟な発想が生まれ試される循環が連鎖的に起こり、継続的に社会が活性化され改革されていくことにある。

デジタル改革で大切なのは、楽観的なチャレンジ精神である。いろいろ試して上手く行くものを探ることが成功への道であり、失敗は糧である。イノベーションは試行の自由から生まれる。そして、ほとんどのチャレンジは環境の変化に適応するために行われる。それまでの前提条件を見直し、創造的なアイデアを生み出す機会となる。つまり、環境変化に応じてさまざまなチャレンジを行うことで、変化に強い社会を作ると同時にイノベーションにも繋がる。

それにもかかわらず、今の日本の社会全体に変化への抵抗感が強まっていることが懸念される。長引く経済停滞の影響もあってか過去の成功体験から脱却できず、内向き思考や組織のサイロ化が進んで、他者の失敗を許容しない、自らはチャレンジしない風潮がある。メディアの報道、テレビのワイドショーやネットのSNSでも批判的な態度とそれに同調する意見、ネガティブな過剰反応が目につく。行政機関は批判を恐れてオーバーコンプライアンスに陥っている。政策においても、変化に慎重な壁に阻まれて進展しない課題が山積している。ビジネスの世界では、環境が変化しているのに先手を打ってチャレンジしなければ衰退するしかないのは常識となっている。本当に守るべきものは風雪に耐えて残る。大切にすべきものこそ試練に晒して洗練させるべきだ。

行政のコロナ対応への批判は真摯に受け止め、改善に努めるべきだ。多くの批判が寄せられて、硬直的な調達制度やプロジェクト管理体制の問題点が洗い出された。一方で、上手くいった対応も少なからずあった。しかしこれらはあまり話題にならなかった。なんらかのチャレンジができて、多少不具合があっても、評価すべきところは評価しなければならない。最悪なのは、失敗を恐れて無策に終わることだ。

日本は世界の中で安全かつ精度のある安心な社会を築いてきたことには定評がある。もっと自信をもつべきだ。デジタル改革は、社会全体が希望的な楽観思考になり、多様性とチャレンジと変化が重要という価値観を共有して進めるべきだ。変われなかった日本社会を変えるチャンスでもあるので、積極的に取り組む歴史的なタイミングだと思う。

本稿では、迅速かつ柔軟に変化に対応しながら創造と進化を繰り返すインターネットの技術の発展を振り返り、社会システムとの関連について考察する。デジタル改革を進めるに当たって、まず、デジタル技術で発展のプロセスが大きく変わってきていて、それに伴い優先順位や価値観も変わっているという認識を社会全体で共有する必要がある。デジタル文化は自由と創造に富んで、変化にも強い。しかし、自然にそうなった訳ではない。インターネット発展におけるアーキテクチャとインターネット文化の役割を通して、仕組みと価値観の共有、その維持発展を支えるコミュニティと継続的な努力の存在を紹介する。デジタルのメリットを生かした社会を構築するにも、自由、創造、チャレンジ、適応など価値観を社会全体で共有し、デジタル的発展を支える仕組みを作っていくこと、そして、仕組みも価値観も軌道修正を繰り返しながら前に進むことを、技術や制度などの仕組み作りに関わる人達と共有したい。

2. なぜデジタル技術が変化に強いのか

デジタル情報はあらゆる物を数値で表し、コピーや伝送で劣化しない。ハードウェアで高速に、ソフトウェアで自由に情報処理が可能で、ネットワークを使ってどんな情報でも交換できる。デジタル技術で技術の発展プロセスが大きく変わった。以前はひとつの技術が比較的独立して発展する「進歩」だったが、デジタル技術ではたくさんの技術が互いに影響し合いながら発展するので、生物の「進化」に似ている。生物の進化は、突然変異と交配による形質変化が多様性を生み、それらが相互依存しながら自然選択を繰り返し環境に適応する。技術も、デジタル化によってコピー、改変、組み合わせのコストが劇的に下がった結果、少し違ったモノが生まれて広まることを繰り返すことで社会環境に適応していると見ることができる。進歩は計画的で予測できるが、進化は複雑で予測が困難だ。なので、IT 技術の未来予測はほとんど当たらないこともうなずける。

技術の世代交代のスピードも飛躍的に速くなった。従来の工業製品の製品寿命は 10 年ほどだが、IT 機器は半導体の機能向上と価格低下で 5 年も経たずに陳腐化してしまう。ソフトウェアの場合、オンラインのアップデートでどんどん機能強化できる。スマートフォンのアプリには毎週のようにアップデートされるものもある。しかも、アップデートにかかるコストはハードウェアに比べて桁違いに小さい。さらに、Web サービスでは、ユーザが気づかないうちに 1 日に何度もクラウド上のソフトウェアをアップデートできるようになった。環境に適応する進化のサイクルがこれまでの技術と比較にならない。

変化に適応するためには、更新することを前提とした設計が必要になる。ソフトウェアの不具合や脆弱性を完全に失くすことは不可能なので、問題が見つければ速やかに修正しなければならない。アプリケーションやサービスの更新は、運用しながら問題点を修正し、同時に機能拡張ができる仕組みである。そうすると設計時の優先事項が変わってくる。今必ずしも必要のない機能は必要性が明確になった時点で作る方が良いものができる。サービスの初期段階では、それよりも将来の発展を見越した拡張性の確保や運用面への配慮が大切となる。

Web サービスでは、開発にもまして運用が重要になってきている。サービスの利用状況をモニタし、また、ユーザから直接フィードバックも得られるので、継続的に課題や問題点を見つけて修正を繰り返すことが可能になった。その結果、進化をドライブするのは運用からのフィードバックになってきている。Web サービスでは、単に作ったサービスを維持、保守するだけでなく、運用から得られたデータを元に機能強化、信頼性向上を行い顧客満足度に繋げる戦略的な攻めの運用が欠かせない。そのためには、運用現場がトラブル対応で疲弊して改善に手が回らない状況にならないように、常に運用の効率化を図ってスリム化し、業務の優先順位を戦略的に考える体制を整えておく必要がある。

開発手法についても進歩型の開発から進化型の開発への変化が起こっている。従来のソフトウェア開発は、長期的な全体計画から中短期計画へと順序立ててブレイクダウンして行くウォーターフォール型開発が主流だった。最近の Web サービスなどでは、開発途中に起こる様々な変化に適応できるように、また、運用からのフィードバックを取り込んで軌道修正できるように、設計と運用を一体化し短い開発サイクルを繰り返しながら計画を進めるアジャイル型開発へと移行している。

さらに、サービスを発展させるのに大切なのは、使われるに従って品質を向上させる仕

組みを作り込むことである。例えば、IT化でユーザからの評価を収集し易くなるが、ユーザ評価を有効に活用するためには、そのための仕組みを作っておく必要がある。つまり、ユーザの要望を素早くサービスに反映したり、ユーザの評価が新規ユーザの獲得やサービス従事者のモチベーション向上に繋がるような仕組みをシステム化しておくべきだ。逆に、折角ユーザ評価を集めても、それが反映される仕組みが無ければサービスは良くならない。より重要なのは、サービスから得られた様々なデータを活用して、サービスの改善に使うことだ。成功しているITサービスは必ず独自のフィードバックの仕組みを持っていて、ユーザからの支持を得て市場で優位に立つことができている。

環境変化に適応するために必要なのは、開発、品質管理、運用の統合による継続的なプロセス改善の実現である。これは単に既存のプロセスをITで置き換えるのではなく、仕事のやり方や組織文化の根本的な変化を意味する。

3. インターネットの文化

インターネットの成功は個々の技術よりも、草の根的な技術の発展と普及を可能にした設計思想とそれを育てた文化によるところが大きい。にもかかわらず、技術の普及ほどには、インターネットを支える考え方や文化が十分伝わっていないのが残念である。インターネットもコンピュータも1970年ごろから急速かつ複雑な発展を遂げる。それを最初に主導したのは政府や大企業だったが、やがて大学や研究機関、そして民間へと主軸が移っていった。テクノロジーと文化は、ハードウェア、ソフトウェア、民生製品における大規模かつ広範囲の試行錯誤で発展したと言える。

パケット交換ネットワークを実証したARPANETの成功を受けて、ネットワークのネットワークを作るプロジェクトが1970年代中頃から米国防総省のDARPAの支援で始まる。1980年代に入ると大学を中心に広がり始め、1985年に全米の大学や研究機関をインターネットで接続するNSFNet計画が始動、NSFNetを中心にインターネットが急拡大するようになる。NSFNetは、当初からの計画通りインターネットの商用化を受けて順次民営化され、1995年に終了する。その間、インターネットを牽引したのは大学と研究機関だった[1,2]。

インターネットの文化として、オープンな共有思考やコンセンサスによる意思決定がよく知られている。このような文化が形成されていった過程も複雑な相互作用の連鎖だったのでいろいろな見方ができるが、本稿では変化に強い仕組みという視点で捉えてみる。

初期のインターネットの文化は、ARPANETのソフトウェアや運用を担った若手研究者が中心となって1970-80年代にかけてできていく。当時の米国の大学には、学生運動やカウンターカルチャーの影響を受けて、既存の体制を疑い、自由を掲げる若者文化があった。

この頃に、コンピュータ文化にも大きな変化が訪れる。1960年代後半にタイムシェアリングシステムが普及し始めると、それまでのバッチ処理に対して、対話型のプログラミングが可能になる。プログラミングをコーディング、テスト、修正の繰り返しでできるようになった。これは、プログラミング作業が、今の言葉で言うウォーターフォール型からアジャイル型になったパラダイムシフトだった。アイデアをすぐに試せる対話型プログラミングに魅了された人達から、オープンで創造的なプログラマの文化が生まれる。注目すべきは、なにか問題が起こった時に臨機応変にプログラムを改変して解決する能力を持つ人がハッカーと呼ばれ高く評価されていたことだ。そして、彼らに絶大な支持を得たのが、プログラマ

がプログラマのために作った UNIX であり C 言語だった[3]。

インターネットもコンピュータサイエンスの研究者たちが自分たちのために作ったネットワークだった。それが次第に研究者以外の人たちからも支持されていく。インターネットが広がったのは、TCP/IP が BSD UNIX に実装されて広く大学で使われたことが要因に挙げられる。シンプルで機種を問わず動作する UNIX が 1970 年代後半から大学で使われ始める。1980 年ごろには UUCP 通信を使ったダイヤルアップ接続により、メールとネットニュース配信もできた。その頃、TCP/IP の利用拡大を図っていた DARPA はそのプラットフォームとすべく BSD UNIX を支援する。そして、1983 年に TCP/IP スタックが実装された 4.2 BSD UNIX がリリースされ、北米のコンピュータサイエンス学科で絶大な人気となった。当時はインターネットに繋がっていた大学よりはるかに多くの大学で UNIX が利用されていた。多くの UNIX プログラマがソースコードや情報の共有にネットワークを積極的に利用したことで、TCP/IP やアプリケーションも急速に完成度が上がった。オープンソースだったので、製品にも取り込まれ瞬く間に広がっていった。

なお、日本でも同様に先に 1970 年代後半に一部の大学で UNIX の利用が始まり、そこから 1984 年に UUCP による JUNET がスタート、1989 年に WIDE プロジェクトによってインターネットに接続された[4,5]。

インターネットと並行して、コンピュータの形態や利用方法も変わってくる。タイムシェアリングは高価な大型コンピュータを複数のユーザが共有する仕組みだった。半導体技術が急速に発展して、集積回路、LSI、マイクロプロセッサが登場し、コンピュータもミニコンピュータ、UNIX ワークステーション、パーソナルコンピュータへとシフトするに従い、計算だけでなく、コミュニケーション、ドキュメント作成、グラフィックス、写真加工から音楽制作までさまざまな創作活動に使われるようになって行く。

1990 年代に入るとインターネットが商用化され、まもなく Web が登場、TCP/IP を実装した Windows95 が発売されると、個人ユーザに爆発的に広まった。いろいろなアプリケーションが動作する PC が、どんな情報でも交換できるインターネットと出会い、誰もが自由に情報を発信できるようになった。PC とインターネットは、創造のための汎用の道具であり、車の両輪となった。新しいアプリケーションが出てくる土壌を与え、多様なアプリケーションを育むことによって、系の進化を支えた[6]。ほとんどの創造は既存のアイデアの（意外な）組み合わせから生まれる。したがって、情報を見つけ、自由に意見を交換し、簡単に試し、経験を共有することが創造に繋がり、これらの相互作用で予想もしなかった発展を見せる。

先述のように、1970-80 年代のコンピュータサイエンスには極めてリベラルでオープンな独自の研究者文化があった。研究者にとってアイデアや成果は仲間と共有するものだった。そのなかでも、インターネットの若手研究者たちには、異なるコンピュータ、異なるネットワーク、異なる組織を繋ぐ必要から、オープンな技術をオープンなプロセスで開発する使命と価値観を共有することが欠かせなかった。初期のインターネットの目的は、組織ごとにばらばらだったネットワークを繋ぐことで、高価なコンピュータリソースの共有にとどまらず、研究者の情報共有やメールなどによるコミュニケーション促進が挙げられていて、人とコンピュータ、そして人と人を繋ぐことだった。

インターネットの普及に伴い、その技術を支える仕組みや組織も変わってきている。初期のインターネットでは、少数のリーダー達がステアリンググループを作って意思決定を行っていた。1986年にインターネットの技術、運用、管理のコーディネーションをするために IETF ができる。最初の会合の参加者数は 20 名あまりだったが、その後急拡大し、翌 1987 年には 100 名を超え、1992 年に 500 名を、1994 年には 1,000 名を超えている。会合への参加者数の増加からもインターネット技術への関心の高まりが窺える。1992 年には、財政面や組織構造強化のため、ISOC が国際非営利組織として設立され、IETF などはその下部組織に組み込まれた。1998 年にはドメイン名や IP アドレスなどの資源管理をグローバルに調整する ICANN が設立されている。インターネットに国境はないので、世界全体のことをグローバルに考える。各国の代表が自国の利益を優先して議論するインターナショナルに対して、グローバルという言葉が好んで用いられる。

IETF はインターネットの標準化を行う任意団体である。IETF はそれまでの技術標準化団体と比べるとかなり異質と言える。標準ドキュメント、メーリングリストでの議論などの記録、参加はすべてオープンとなっている。参加者は個人の資格で参加し発言する。IETF のモットーとして有名な「rough consensus and running code」は、投票ではなくコンセンサスによる意思決定プロセスと、何より実際に動いているコードを信用するという独自の文化をよく表している。つまり、他の標準化団体では、投票制度が往々にして票取りの駆け引きや拒否権の発動に繋がってしまうこと、実装からかけ離れた複雑な標準ができることを批判しながら、IETF の技術者による実用的な標準化プロセスのあるべき形を示している。実は 1992 年にこの言葉が登場する背景には IETF の危機があった。インターネットが急拡大するなかで、技術的にはアドレスの枯渇や経路表の増大、外部的には OSI 標準とそれを支持する既存勢力との標準をめぐる競争があり、内部的にも OSI との協調や意思決定プロセスに対する不満が出てきて IETF は分裂の危機にあった。この IETF スタイルが言語化されたことで、IETF はまとまりを取り戻し、インターネットが標準としての支持を広げていくことになる[7,8]。当時の IETF には、自分たちが技術で未来を作っているという熱気が溢れていた。

ちなみに、インターネットの技術標準は、RFC (Request For Comments) というドキュメントである。RFC は、まだ ARPANET 時代の 1969 年に、ARPANET 開発運用に携わっていた若手研究者たちが、プロジェクトのメンバー以外とも自由に意見を交換するために非公式な技術メモとしてスタートした。この技術情報の回覧方法が情報共有に有益だったので、技術標準の規定にも RFC を使うようになり、今日に至ってもコメント募集という形式を維持している。現在も、RFC は標準以外の情報共有のためのドキュメントや一部にはエープリルフールのジョークも含んでいて、標準は RFC の一部にすぎない。このあたりにもインターネット文化らしさが出ている。

4. インターネットとアーキテクチャの役割

インターネットから数知れないイノベーションが生まれてきた。誰もがアイデアを簡単に試して、サービスを始めることもできる。実際に、Google も Facebook も学生が手作りでサービスを実装して始め、やがて巨大企業に成長することになる。これを可能にしたのは、自由に機能を端末側で導入できるようにしたアーキテクチャである。

4. 1. アーキテクチャの役割

コンピュータシステムにおけるアーキテクチャは、複雑なシステムを抽象化し、その基本構成を示し、各構成要素の役割や機能を明示する。また、構成要素の境界における接続仕様であるインターフェイスを示す。アーキテクチャは、システム設計の物差しであり、機能の分かりやすい分離や内部構造の隠蔽などの役割がある。抽象化にはモジュール化や階層化が使われる。その際、どの部分を共通化あるいは固定して、どの部分を自由にするかの線引きを行う。それぞれの要素技術が自由に発展するために、各要素技術の役割を明確に規定し、外部とのインターフェイスを定め内部をブラックボックス化することによって、各要素技術が独自に内部を改変できるようにするのだ。これは、自律した異なるネットワークを繋ぐインターネットを考える上で根源的問題だった。

このように、技術の発展を支える基盤となるアーキテクチャは、構成要素の抽象化と境界を定めるインターフェイスの設計によって、システムが発展していく形を規定し、ひいては産業の形を決めることにもなる。インターネットの階層構造がルータやスイッチなどのネットワーク機器やソフトウェアの発展の仕方を方向づけた。大学や研究機関の自律したネットワークを相互接続するアーキテクチャがそのまま商用化されて、ISP の相互接続とトランジットやピアリングという接続形態に発展した。ネットワークはアプリケーションに関知しないアーキテクチャが、多様なサービスの展開を促進した。このように、基盤技術のアーキテクチャは将来の技術発展に大きな影響を与える。基盤技術を作ることは、将来実現されるであろう便利を妄想することであり、発展の制約となりうる要素を排除することでもある。

インターネットのアーキテクチャもいくつかの切り口があるが、根本にあるのはネットワークのネットワークだと言える。最初から構想されたのが、自律した独自技術を使ったネットワーク同士を、中央がない分散システムとして相互接続していくためのオープンなアーキテクチャだった。これを実現するために、ネットワークの階層構造として、ベストエフォートでパケット配送を行う IP と、エンドツーエンドでバーチャルサーキットを実現する TCP が分離された。IP 層だけを共通化することで、多様な通信技術とアプリケーションをサポートできるアーキテクチャとなった。ネットワーク上の識別子にはネットワークの接続点を示す IP アドレスが規定され、ネットワーク間を接続するゲートウェイ（後にルータと改名される）の役割が定められた[9,10,11]。

4. 2. 設計原理の重要性とエンドツーエンドの原則

インターネットが発展していくうえで、システムの構造を示すアーキテクチャと共に、その設計原理が重要な役割を果たしてきた。設計原理が、構成要素を設計する際にもどのような選択をすべきかを導き、全体の文化を形づくることになる[12]。

まず、挙げられる設計原理はベストエフォートである。ベストエフォートはもともとパケット配送の信頼性を表す概念だったが、それを遥かに凌ぐ普遍性をもつ考え方になった。「完璧でなくてもいい、できるだけ頑張る、できない時はゴメンする、まわりがカバーするから大丈夫」と言うもので、楽観的であり論理的に聞こえない方式だ。しかし、このような方式で通信が上手く動くことを理論と実装で示して専門家を驚かせたのがイーサネット

の登場だった。ベストエフォートという言葉は、1973年のイーサネットの論文で提唱されている[13,14]。インターネットのパケット配送もベストエフォートだ。この考え方をを使うと、信頼性の低い部品を集めて高信頼システムが作れる。信頼性を単体のコンポーネントで上げようとするるとどんどんコストが高くなるが、システム全体で考えれば効率の良い役割分担を実現できる。さらに、一部が壊れても動き続けるように構成できるので、災害にも強いシステムも作ることができる。

インターネットの設計原理のなかでも、その後の技術発展において重要な役割を果たしたのがエンドツーエンドの原則[15]である。ネットワークはベストエフォートなパケット配送だけを提供し、インテリジェントな機能は端末側に持たせるべきだとし、ユーザが新しいアプリケーションを自由に導入できるようにした。賢いネットワークを提唱していた中央集権の電話網とは真逆の設計だったが、これこそがその後の空前の技術発展に繋がった[16,17]。

エンドツーエンドの原則は、通信路上に伝送誤りや故障の可能性がある以上、エンドシステム間でのデータの検証が必要だという認識から生まれた。例えば、ファイル転送の場合、いくら信頼性の高い伝送路を構築しても、それだけでは転送を保証できない。他にも、ルータやホストにおけるメモリのビットエラー、ファイルシステムのバグ、ディスクのエラー等が起こり得るので、最終的にはアプリケーションがエラーチェックをする必要がある。さらに、もし、ネットワーク内部に特定のアプリケーションを想定した機能を組み込むと、他のアプリケーションにとっては制約となる可能性がある。例えば、再送機能を組み込んでしまうと、音声通話などでは不必要な再送による遅延変動が問題になる。将来登場するアプリケーションを予測できない以上、その発展を阻害することになりかねない。

エンドツーエンドの原則は、ネットワークをできるだけシンプルにして、機能の追加や変更は端末側で自由にできるようにするものである。しかし、最初からこのように明確に考えられていた訳ではない。1980年ごろから議論を重ねて徐々に明確かつ広範な設計原理となった。始めはエラー修正をどこで行うかという純粋な技術論だったが、次第にインターネットを特徴づける設計原理になっていく[17,18]。

エンドツーエンドの原則をめぐる議論は最近でも繰り返されている。その代表的なものはネット中立性の議論である。これは、通信事業者が特定のサービスやコンテンツを制限あるいは優遇すると、ロックイン効果で自由にサービスが生まれ育つインターネットの良さを損なうので、そのような差別化を規制すべきだとする意見である [17,19]。これはまた、2018年の海賊版漫画サイトのブロッキング議論[20]における反対根拠でもあった。

4. 3. エンドツーエンドと技術の発展

エンドツーエンドの原則から多様なアプリケーションとビジネスモデルが登場した。当初のアプリケーションはメールとファイル転送ぐらいしかなかったが、その後、Web、検索サービス、ブログ、eコマース、ネットオークション、ソーシャルネットワーク、電子書籍、音楽配信、動画配信、アプリストア、ビデオ会議、地図サービス、乗り換え案内、シェアリングエコノミーなどなど、すべてインターネットから生まれてきた。

技術は単に技術面が優れているだけでは普及しない。成功した技術には経済的なインセンティブが組み込まれている。Web が急速に普及したのは、ユーザが無償のブラウザをイ

インストールするだけで簡単に利用可能だったことが大きい。それに加えて、初期に Web クッキーが導入されたことで利用方法が広がり、商用利用に道を開いた。Web のプロトコルはサーバに状態を持たないシンプルなリクエスト・レスポンスなので、連続したリクエストを関連づけることができず、そのままでは対話的なやりとりができない。そこで、最初のリクエスト時にクッキーと呼ぶ識別子をブラウザに渡し、2回目以降のリクエスト時にクッキーを提示してもらうことでブラウザを特定できるようにしたことで、対話的な買い物などのやりとりができるようになった。

検索エンジンが成功したのも、たくさんのユーザが使うほどに検索結果の品質が向上するフィードバックの仕組みと、ユーザの志向に合ったオンライン広告を出したい広告主とを組み合わせるビジネスモデルが上手くいったからである。e コマースは、リアル店舗では扱い難いニッチなロングテール商品と、ユーザのオンライン行動履歴から「おすすめ」を行いマーケティングに繋げることに成功した。利用拡大と頻繁なアップデートで品質向上した例として、コロナ禍で急速に普及した Zoom が記憶に新しい。Zoom アプリはコロナ禍でほぼ毎週のようにアップデートされていて、多くの機能が追加され、格段に使いやすくなった。

また、予想外の技術進化の例として、HTTP ストリーミングによるビデオ配信がある。安定した通信が期待できないインターネットでは、ビデオが途切れないようにリアルタイムでデータを継続して配信するのは難しい。そこで、初期のビデオ配信では、転送時間の読めない TCP の代わりに、不安定な通信環境でもできるだけ安定してデータを送る工夫を凝らした独自プロトコルの開発と利用が盛んに行われた。そのため、ライブビデオ配信には専用のサーバと専用のプレイヤーを使うのが常識だった。その一方で、オンデマンドなら全体をダウンロードしてから再生していた。そのうちに、画像転送に使われていたプログレッシブダウンロード方式を音楽ファイルの再生に応用した手法が、低解像度のビデオ再生にも使われるようになり、それがさらに発展して、ビデオを数秒分のファイルに分割して HTTP で連続ダウンロードしながら再生する HTTP ストリーミングが登場する。この方式は、既存の HTTP の仕組みを使っているので専用装置や専用サーバが不要で、大規模配信にも既存の Web 用設備が使える経済性が大きな利点だった。今では、ネットワークと PC の性能向上に伴って、高解像度のビデオにもライブ配信にも HTTP ストリーミングが使われるようになってきている。

一方で、アーキテクチャで固定されたために、後で変えることが難しくなった技術もある。その代表例は、IPv4 アドレスだろう。IPv4 アドレスは約 40 億の空間で、世界の人口にも足りない。IPv4 のさらに 40 億倍のアドレス空間を持つ IPv6 への移行は、アドレス変更以外の要因もいろいろあるにせよ、30 年もかかってもまだ終わりそうもない。他にも、技術は確立されたが、ネットワーク内部の変更が必要だったり、あるいは経済的なインセンティブの不足で普及に至らなかった技術として、IP マルチキャスト、QoS 制御、モバイル IP などが挙げられる。

セキュリティ技術にもあまりインセンティブが働かなかった。初期のインターネットは、研究者が自分たちで使うために作ったので性善説に基づいていて、悪用対策が不十分だった。今やインターネットの利用者数が世界人口の半数を超え、利用者の前提が根本的に変わった。乱用や悪用が増えて、その手口は巧妙になり、大きな問題となっている。初期のイン

ターネットでは、すべてのプロセスがオープンで透明化されていたので、不正利用は難しく対策の必要性も低かった。その後、商用利用が増え続け、知的財産保護やプライバシーへの配慮で透明性も低下したにもかかわらず、セキュリティ対策は後付けになったまま今日に至っている。今後に向けて、人だけでなく IoT や AI も対象に含んだセキュリティの概念を構築することが課題となっている。

5. 変化に強い仕組み

5. 1. 対話型チャレンジと失敗の奨励

これまで見てきたように、デジタル技術で環境変化に対応するプロセスが大きく変わった。アイデアを試し結果を見て修正することを繰り返す、対話型チャレンジができるようになった。自由なチャレンジが創造とイノベーションを生み出し、変化に強い社会を作る。これは、多様性を育む、弱者に優しい社会でもある。計画に従うことより変化への対応に価値を置き、常に変化する社会および技術に適応し続けるようにならなければならない。

オープンな文化と透明性のあるプロセスが自由な発展を促進する。アーキテクチャが、変わらない部分と変えていい部分の線引きをする。その設計原理がブレない芯となり、自由な発展の方向性を示す。これらを支える土壌になるのは、自由で制約の少ない社会である。環境変化で生じている問題や不便を発見して共有する、改善のアイデアを交換し試してみる、結果と学びを共有する、これらのサイクルを回す、いずれも自由にできることが欠かせない。

シリコンバレーには、技術で世の中を変えようとする人達が集まり、独自のシリコンバレーの文化を作っている。そのなかに「fail fast, fail often」と言う失敗の奨励がある。挑戦した結果の失敗は学びの宝庫であり、善である。逆に、失敗しないようにすることは、失敗に対応する能力開発をしないことを意味する。デジタル技術は新陳代謝が速い。たとえサービスが人気になっても激しい競争が待っている。常にベストなサービスを提供するよう改革を続けなければ負けてしまう。さらに、新しいビジネスモデルが登場して時代に取り残されるかもしれない。自己改革ができなければ未来はない。デジタル社会の形成にも、チャレンジを推奨し、失敗を受け入れる文化を醸造することが大切である。

5. 2. 標準化の役割

標準化の役割のひとつに、誰もが市場参入できるようにすることがある。複雑なシステムはベンダーロックインを招きやすい。環境変動に適応して進化するシステムには標準化の努力が欠かせない。標準化は、異なる製品や実装が相互に繋がり動作するためのインターフェイスの共通仕様を定める。標準化によって、同じ機能を持つ製品が複数のベンダーから提供可能になり、特定のベンダーによるロックインを防ぐことができる。

しかし、標準化だけでは不十分である。インターフェイスはアーキテクチャがあってはじめて定まり、標準化できる。したがって、アーキテクチャが確立されていない技術を標準化する場合は、まずアーキテクチャに取り組み、設計思想を明確にする必要がある。この部分は、エンジニアリングよりアカデミズムの領域となる。実際、米国国立標準技術研究所（NIST）には多くの研究者がいて、計測や技術標準の策定と並行してアーキテクチャドキュメントの作成に力を入れている。EU では、欧州電気通信標準化機構（ESTI）が同様の役

割を担っている。そして、定められた標準が、本当に標準としての役割を果たすためには、適合性をテストし、インターオペラビリティを確認する作業も欠かせない。

5. 3. ロバストネス

自由や多様性にはマイナスも伴う。実際、インターネットでも自由と多様性を容認するアーキテクチャが乱用され、迷惑メール、マルウェア、フェイクニュースなどがはびこる原因となっている。しかし、問題が起こること自体は必ずしも悪いことではない。問題の発見はチャレンジの種である。また、問題に対応する繰り返しのなかから新しい技術や秩序が生まれてくる。問題部分も含めて多様性を許容し、発生する問題への対応を繰り返すことで、問題解決機能を維持し、システムのロバストネスを支えることができる。その意味で、次々に問題が起こり、それらを乗り越えていくことは、創造と進化にとって欠かせないプロセスだと言える。また、ここで養われるバランス感覚が環境変動に対応する際に重要な役割を果たしている。

インターネットのセキュリティの問題についても同様のことが言えるだろう。最初から悪用防止を優先していたら、自由な発展もなかったと思われる。バランスの取り方は難しく、正解はない。全体を見れば性善説でシステムを作ることは間違っていないだろうが、長い目で見れば悪用が得にならないように歯止めの制度を設けるなど、技術と制度を組み合わせた解決を探るべきだと考える。そして、技術と制度を繋ぐために、まず必要なのが ID とトラストの設計だ。現実空間の「ひと」を仮想空間にどう関係づけ、そのためにどのような ID とトラストの仕組みが必要かを明確にすることが課題だと考えている [21]。

5. 4. キープイットシンプル

チャレンジは大切だが、もちろんむやみにチャレンジすれば良い訳ではない。成功の確率を上げるための手法はいろいろあり、効率よくチャレンジするべきだ。一方で、効率化しすぎてはいけない。複雑な進化では思わぬものが新しい環境に適応するために役に立つ。多様性を育むには多少のムダを許容する余裕が必要となる。

ムダの許容は、実は多くの技術と相反する。一般にほとんどの技術は、想定した環境における効率化や最適化をするものだ。それゆえ、現状の利用形態に最適化された技術は、環境変動で時代遅れになる。デジタル技術は環境変化が速いので、長期的に見れば、特定のサービスのための専用機や専用システムは生き残れない宿命にある。特に、例外処理や最適化技術は短期解だと考えるべきだ。同様に、危機対応用に作られたシステムは使う機会が少ないため世代交代が遅く、なかなか完成度が上がらず、また、すぐに陳腐化してしまう。普段使っていないものはいざと言う時に使えない。危機対応用に専用システムを作るのではなく、普段使うシステムを危機にも対応できるようにするべきだ。

多くの技術的問題は、長期的に見れば一過性のものであり、技術が発展するにつれ、その解決策も必要がなくなる。もちろん直面する問題を解決する技術は有用である。例えば、ゲーム端末は操作性、処理速度、価格面のバランスで汎用 PC に優る。しかし、PC の性能も向上していくので前世代のゲームには十分な性能を持つようになる。有益な技術も環境が変わると不要になり、場合によっては、進化を阻害することになる。特に基盤技術では、最適化を導入する場合でも、将来その技術が不要になれば速やかに取り除き、進化を阻害しな

いようにする考慮が必要だ。

システムは次第に機能追加されて複雑化する宿命にある。だが、複雑化すると変化に対応できなくなる。さらに、時代遅れになったレガシーシステムを使い続けると、その維持コストは次第に膨らんでいき、やがて膨大なものになる。したがって、システムを常に見直し、スリム化効率化をはかる努力を怠ってはいけない。これは社会制度や規制にも当てはまる。不要になったルールは廃止し、規制は緩和していかないと活力が失われる。デジタル化にあたってまずやらないといけないのはこれらの見直しだろう。

5. 5. 制度と知財

インターネットは、ある意味で既存の制度に囚われなくて、あるいは疑問を呈するようにならされてきた。その発展の途中で、既存の法制度や慣習と摩擦を起こしてきている。その代表的なものは、オープンと知財のバランス問題である。

オープンソースを例に挙げると、この言葉ができる前から、研究者間ではプログラムのソースコードは共有するものだった。プログラミングは巨人の肩に乗って行う作業であり、既存のコードを利用して必要に応じて改変するのが当たり前だった。つまり、他人のコードを見て学び、利用させてもらい、自分のコードもまた誰かの役に立てば幸いだと考えられていた。1970-80年代にソフトウェアが商用化され知財化されていくと、ビジネスのために知財を守ろうとする派と、以前からのオープンなソースコードの共有文化を守ろうとする派に二極化して度々衝突することになる。UNIXも巻き込まれ、そのゴタゴタの中からLinuxが誕生することになる[22]。いまではオープンソースの価値は広く認知されるようになった。参照実装の役割としてオープンソースは必須である。オープンソースがなければプログラマは育たないし、実装の完成度も上がらない。

規制によって技術が発展しなかったのがセキュリティ関連技術だ。軍事機密という名目のもとで研究開発や情報開示が制約されてしまった。特に、基本技術である暗号技術は米国の暗号輸出規制の対象で、2000年までは暗号技術の利用や公開は大きく制限されていた。今でも規制が撤廃された訳ではなく、国ごとに複雑なカテゴリ分けと規制が存在する。暗号関連のソフトウェアの公開が差し止められたり、開発者が訴訟に巻き込まれた。研究者たちは暗号技術を使いたくても使えなかった。

著作権や特許は、もともとは創作を促進するための制度だったが、イノベーションの進化スピードに制度が追いつかないのが現状だ。相互作用で連鎖的に技術が発展するようになると権利者の特定すら難しく、これまでの独占権が当てはまらない事例が増えている。現在では、イノベーションの促進のために作られた制度が逆に阻害要因になっていることの方が多くなっている。デジタル社会に向けて、制度の改革が課題となっている[23]。

6. デジタル社会の実現に向けて

日本の社会システムの再構築、デジタル化は待ったなしの状況だ。デジタルの機動性と創造力を生かすためには、自由と創造、適応と挑戦の価値観を社会で共有する必要がある。基盤となる制度やシステムには、デジタル技術によって発展を支える仕組みが必要であり、そのための全体的なアーキテクチャの設計が肝要となる。発展を支える構造を抽象化し、理念を言語化し共有し、さらに、それらを運用してメンテナンスし続けることにより持続的に

発展するシステムが確立する。その上で、このようなデジタルシステムのプロセスに整合する制度と組織を整備し、計画から適応へを確実に遂行し、ルール型から運用型や裁量型へ転換をはかり、利用者の声を受け止め、評価し、必要な繁栄を迅速に行う仕組みを確立していく必要がある。

事態の推移の多様性を事前に網羅して準備することは完全にはできない。多様性を順次包含していくことはルールだけではカバー出来ない。さらに、すべての人のシステムは、公平だけでなく公正 (equity) を実現する必要がある。単に偏らないで均一に扱うことが平等ではなく、すでに存在する不公正な社会格差の是正に重点をおいた対応が大切だ。柔軟に対応するためには、目的を共有して、現場の裁量を増やす必要がある。人材についても、技術が個別に進歩していた時は、専門に細分化することが効率的だった。しかし、進化を促進するには、異なる分野のアイデアの交流が必須であり、仕事の進め方も変わる。

昨年 (2020 年) に政府のデジタル改革関連法案ワーキンググループが、誰一人取り残さない、人に優しいデジタル化を進めるために、10 項目からなるデジタル社会を形成するための基本原則を公表した。本稿では、この基本原則の「オープン・透明」「公平・倫理」「包摂・多様性」などの価値観を社会で共有し、「安全・安心」「継続・安定・強靱」「迅速・柔軟」を組み込んだ社会システムを構築することで、「浸透」「社会課題の解決」「新たな価値の創造」「飛躍・国際貢献」に繋げることを提案したと関連付けられる。特に、「迅速・柔軟」の内容は、「小さく産んで大きく育てる、デジタルならではのスピード化の実現」、「社会状況やニーズの変化に柔軟に対応できる制度・システム」、「アジャイル発想を活用し、費用を抑えつつ高い成果を実現」、「構想・設計段階から重要な価値を考慮しアーキテクチャに組み込む」となっている。本稿ではその背後にある考え方を、変化に対応するという切り口から述べた。

本年 (2021 年) に発足したデジタル庁は、これまで同じような機能が縦割りでバラバラに開発されているものを統合し横串を通すことを目指している。社会が多様な現場のニーズに対応しながら発展するためには、共通項を括り出したサービスを現場が必要に応じて自由に改変、それが広く共有され、良いものが残っていくアーキテクチャを作り込み、その理念を浸透させ文化にしていくことが大切である。

最近では、他の行政機関においても新しい取り組みが始まっている。アーキテクチャの考え方をより広く捉えて応用しようとする IPA デジタルアーキテクチャ・デザインセンターの産業アーキテクチャへの取り組み[24]や、アジャイル開発手法をビジネスや政策にも適用する経済産業省のアジャイルガバナンス提言 [25,26]などの活動にも期待したい。

最後に、教育について触れておきたい。コロナ禍で大きく影響を受けたのは教育の現場だ。我々も長らく高等教育のオンライン化に取り組んできたが、初等教育で対面授業ができなくなるための対応は充分でなかった。デジタル社会の創生で教育に対する役割と責任は大きい。人類全体が直面していることとして重要なのは、デジタル環境で「知と学び」の本質が変わってきたということだろう。いまや知識はネット上にあり、学びのための教材はオンラインでも提供される。人の成長が人から学ぶことであることから、人類すべてが瞬時につながるデジタル文明を前提に、理想の実現や課題の解決に取り組む次世代への環境の整備こそが、技術と技術政策にかかわるすべての者の使命である。

参考文献

- [1] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, and Stephen Wolff. A Brief History of the Internet. *ACM SIGCOMM Computer Communication Review* 39(5):22-31. October 2009.
- [2] Peter J. Denning, Anthony Hearn, and C. William Kern. History and overview of CSNET. *ACM SIGCOMM '83*. April 1983.
- [3] Dennis M. Ritchie and Ken Thompson. The UNIX time-sharing system. *ACM SOSP'73*. January 1973.
- [4] WIDE プロジェクト. 日本でインターネットはどのように創られたのか? WIDE プロジェクト 20 年の挑戦の記録. インプレス R&D. 2009.
- [5] 村井 純. インターネットの基礎 情報革命を支えるインフラストラクチャー. KADOKAWA. 2014.
- [6] Jonathan L. Zittrain, *The Future of the Internet -- And How to Stop It*. Yale University Press & Penguin UK. April 2008. ジョナサン・ジットレイン. インターネットが死ぬ日: そして、それを避けるには. 早川書房. 2009.
- [7] Andrew. L. Russell. 'Rough Consensus and Running Code' and the Internet-OSI Standards War. *IEEE Annals of the History of Computing*, 28(3) 48-61. August 2006.
- [8] Geoff Huston. Running Code. *The ISP Column*. August 2021.
<https://www.potaroo.net/ispcol/2021-08/runningcode.html>
- [9] Vinton Cerf and Robert Kahn. A Protocol for Packet Network Intercommunication. *IEEE Transactions on Communications*, 22(5) 637-648. May 1974.
- [10] Jonathan B. Postel. Internetwork Protocol Approaches. *IEEE Transactions on Communications* 28(4):604-611. April 1980.
- [11] Leslie Daigle, On the Nature of the Internet, GCIG Paper No. 7. 2015.
https://www.cigionline.org/sites/default/files/gcig_paper_no7.pdf
- [12] Clark, David D. *Designing an Internet*. The MIT Press. October 2018.
- [13] Robert M. Metcalfe. Packet Communication. MIT/LCS MAC TR-114. December 1973. <http://publications.csail.mit.edu/lcs/pubs/pdf/MIT-LCS-TR-114.pdf>
- [14] 喜多 千草. 起源のインターネット. 青土社. 2005.
- [15] J. H. Saltzer, D. P. Reed and D. D. Clark. End-To-End Arguments in System Design. *ACM Transactions on Computer Systems*, 2(4) 277-288. November 1984.
- [16] M. S. Blumenthal and D. D. Clark. Rethinking the design of the Internet: The end to end arguments vs. the brave new world. *ACM Transactions on Internet Technology*, 2001.

- [17] Barbara van Schewick. Internet Architecture and Innovation. The MIT Press. 2010.
- [18] Gillespie, Tarleton. Engineering a Principle: 'End-to-End' in the Design of the Internet. Social Studies of Science, v36 n3, June 2006: 427-457.
- [19] 総務省. ネットワーク中立性に関する研究会中間報告書. April 2019. https://www.soumu.go.jp/main_content/000613654.pdf MIC. Study Group on Network Neutrality Interim Report. https://www.soumu.go.jp/main_content/000666187.pdf
- [20] 総務省. インターネット上の海賊版サイトへのアクセス抑止方策に関する検討会報告書. August 2019. https://www.soumu.go.jp/main_content/000638089.pdf
- [21] 村井 純, 鈴木 茂哉, 松尾 真一郎, クロサカタツヤ. ニューノーマル時代における人間の社会活動を支える情報基盤の在り方と デジタルアイデンティティの位置づけ. August 2020. <https://kbcl.sfc.keio.ac.jp/TR/global-digital-identity-for-new-normal/>
- [22] 藤田 昭人. Unix 考古学 Truth of the Legend. ドワンゴ. 2016.
- [23] Matt Ridley. How Innovation Works: And Why It Flourishes in Freedom. Harper. 2020 マット リドレー. 人類とイノベーション: 世界は「自由」と「失敗」で進化する. ニューズピックス. 2021.
- [24] IPA デジタルアーキテクチャ・デザインセンター. <https://www.ipa.go.jp/dadc/>
- [25] 経済産業省. GOVERNANCE INNOVATION: Society5.0 の実現に向けた法とアーキテクチャのり・デザイン. July 2020. <https://www.meti.go.jp/press/2020/07/20200713001/20200713001-1.pdf>
- [26] 経済産業省. GOVERNANCE INNOVATION Ver.2: アジャイル・ガバナンスのデザインと実装に向けて. July 2021. <https://www.meti.go.jp/press/2021/07/20210730005/20210730005-1.pdf>