

情報システムと経営のアーキテクチャ

小尾敏夫先生早稲田大学ご退職に寄せて

國領二郎[†]

Architectures of Information Systems and Business

Jiro Kokuryo

A CIO's job can be conceived as one of harmonizing various architectures within an organization, such as product architecture, organizational process architecture and information systems architecture. CIOs should predict the future directions of the industry and align various architectures, so they will be mutually reinforcing, rather than contradicting. The task will have to be a dynamic one because architecture change with the changing environment. The change can be either incremental or revolutionary, depending on whether or not the critical bottlenecks within the architecture shifts or remains the same.

1. ICTと経営のアーキテクトとしてのCIO

永年、実務と理論の両面にわたって情報通信産業、特にCIO（Chief Information Officer）の研究をめぐってご活躍下さった小尾敏夫先生に敬意を表したい。その現れとして、ここでは筆者なりに、過去1970年代以来の経営組織と情報技術の変遷と、そこにおけるCIOの役割について振り返ってみる。基本的視座は、情報システムを含む様々な経営をめぐる仕組みのアーキテクチャ設計と、その構築と進化をマネジメントする「アーキテクト」としてのCIOである。情報システムの経営インパクトが高まるにつれて、その仕事は真に経営トップでなければできないものになりつつある。

過去の文献検索などをすると、CIOという概念が経営分野で広がってきたのは1980年代の初期に企業におけるコンピュータ利用が進化し、重要度が高まってきたことに伴っている（Guputa, 1991）。これは情報システムが単なる業務システムの域を超えて、企業の競争戦略そのものを変える基盤となりうる（McFarlan, 1984）という認識が広がったのと軌を一にしている。経営戦略の中核を情報システム戦略が占め、その意思決定を行うのも経営トップ自らが行わなければならないという認識である。小尾先生が情報産業やCIOの役割に関心を持ち始められたのもこの頃だったと理解している。

注意すべきは、この時期に企業の情報システムが大きな変革期にあったことだ。システムのオープン化やネットワーク化、分散化などが進む中で、力点が個々のシステム構築から、企業全体の情報環境の整備に移ってきたのである。システム管理的に言うと、これは情報システムがシステム部門の独占物であったものから、企業内の事業部門がそれぞれにシステムを開発する（あるいは外部に発注する）時代になってきたということである。しかし、それを無計画にやっていると企業バラバラにな

[†] 慶應義塾大学／早稲田大学非常勤講師

るし、戦略的な動きができない。企業全体の情報環境を設計し、整える「アーキテクト」が重要になるゆえんである。

このような認識の流れに大きなインパクトを持ったのが、Hopper氏がハーバードビジネスレビューに発表した論文（Hopper, 1990）である。アメリカン航空に決定的な競争優位をもたらした、SABRE 座席予約システム導入の指揮をとった人物の論文である。その中で Hopper 氏自身は CIO という言葉を使ってはいないが、情報システムの性格が変化する中で、企業にとって個別のシステム構築がもはや課題ではなく、“building electronic platforms that can transform their organizational structures and support new ways of making decisions.” ことが中心となると説いている。しかも、“companies will be technology architects rather than system builders, even for their most critical applications.” と説いて、CIO の役割が企業のシステム構築自体をすることではなく、設計指針を示すものになると予言したのだった。技術的には、IBM が大規模で多種のシステムが混在する環境のために Enterprise System Architecture を提案したのが 1988 年であった（Aken, 1989）ことも、この時代の流れと符合している。

1990 年代に入ってから動きは急だった。特に商業利用が開始されたインターネットのインパクトは極めて大きく、従前は企業内のシステムを中心に考えていればよかった CIO の仕事も、いまは経済全体の情報システムのエコシステムの中で自社の情報基盤を考えなくてはならなくなってきている。

上述の Hopper 論文は、（システムではない）情報そのものが競争力の源泉となる時代が来るとも予言した。これも、クラウドコンピューティングの登場によって本格的になってきた。クラウド上に蓄積されたビッグデータが産業を問わず重要な競争優位の源泉となる時代が到来しており、名実ともに Chief “Information” Officer の時代となっている。

2. アーキテクチャの概念

CIO の仕事を設計者（アーキテクト）と位置づけるからには、アーキテクチャの概念を明確化しなくてはならない。この作業によって、CIO の具体的な姿が見えてくる。

基本となるのは、サイモン以来の人工物の考え方だろう（Simon, 1981）。単純で独立性の高い下位システムの組み合わせで構成される階層構造としてシステムを捉え、その階層構造のあり方をアーキテクチャと捉える考え方である。このような捉え方によって、経営システムと情報システム、さらには製品なども、アーキテクチャという共通の枠組みの中で考えることができる。

ここで、下位システムの独立性の程度と、下位システム間が相互作用する方式（インターフェース）の汎用性という、アーキテクチャの特質に影響を与える 2 つの軸に留意する必要がある。例えば一つの軸である下位システムの独立性が高い場合には、1 つの下位システムの設計変更があっても、他の下位システムの設計変更は小さくてすむ。ただし、独立性を高く保つためには、性能などに大きめのバッファータを持たなければならない。そのような独立性の高いものを本論文ではルース（緩結合）アーキテクチャと呼ぶ。逆に独立性の低いものをインテグラル（密結合）アーキテクチャと呼ぶことにする。下位システム間のバッファータが小さく設計上の相互依存性が高いインテグラルなアーキテクチャにすると、1 つの下位システムの設計を変更した時に、他の下位システムも変更しなければなら

なくなる可能性が高まる。

もう一つの軸はインターフェースのオープン性である。世の中には特定の企業のシステム間のみを結合させるために作られた固有の（クローズドな）インターフェースもあれば、企業をまたがって多様な製品を結合させることを狙ったオープンなインターフェースもある。一般にインテグラルな設計をするものにはクローズドなインターフェースが用いられ、ルーズな設計をするものがオープンなインターフェースを用いる場合が多い。

このあたりの認識が重要なのは、密結合（インテグラル）型のアーキテクチャを採用するか、緩結合（ルーズ）型のアーキテクチャを採用するかということが、企業の製品戦略だけでなく組織戦略にも大きく影響してくるからである（藤本隆宏 et al., 2001）。その意味で、全体をどのような形に構造化し、構造化された下位システム間にどのような相互依存性を持たせるかはアーキテクチャ設計の基本課題である。組織能力的には、インテグラルな製品を作る企業には「すり合わせ」能力が要求され、緩結合型製品を作る企業には、オープンイノベーションを行う能力が要求される。このように、製品設計と組織プロセス（含む情報システム）はバラバラの課題ではなく、整合性をとるべきものである。それを実現することこそ、CIOの最大の使命であるというのが本稿の主要な視座である。

どのようなアーキテクチャが良いかという問題は、インテグラルアーキテクチャが得意とされる日本企業も米国情報産業などに倣ってオープン化（モジュラー化）すべきか、という文脈で盛んに議論された。コンピュータ産業ではオープン化が進んだが一律にオープン化が良いわけではない。たとえば、石油危機の時に日本の自動車メーカーが伸長したことを想起すると、システム全体を徹底的に最適化する必要がある製品の場合にはインテグラルな設計を行うのが一般的に良いと考えることができる。また、同一産業においても技術の進化に伴って、適切なアーキテクチャが変化することにも留意しなければならない。さらには、一時点においてドミナントなアーキテクチャも時代の変化によって陳腐化する。このようにアーキテクチャの設計においては、企業の組織から風土にいたるまで、全てを作り替える必要が生じうることを意味している。CIOが経営トップでなければ務まらないゆえんといえよう。

3. カイゼン（アーキテクチャ内イノベーション）とアーキテクチャ革命

アーキテクチャをイノベーションの文脈で考えた時に、柔軟性をもたらす面と安定性をもたらす面のパラドックスがあることに気づく。アーキテクチャの価値は、歩みの遅い下位システムが足をひっぱることのないように全体設計から分離することで、他の下位システムの自由な進化をうながす柔軟性にあるからだ。すなわち下位システム間の相互作用のインターフェースを固定することによって、いちいちシステム全体の設計をやり直さずとも進化の速い部分に集中することができる。ところがこの柔軟性はシステム全体からみた時に、アーキテクチャやインターフェースそのものは変化させないルールを守るといふ安定性と裏腹の関係になっていることに留意したい。結果として、アーキテクチャそのものが古くなって全体のパフォーマンスが上がらなくなることがあるのだ。

この現象を incremental（漸進的）なイノベーション対 radical（革命的）なイノベーションという対比で考えることもできる。アーキテクチャを安定させるメリットと、アーキテクチャそのものを見直す必要性のバランスにおいて、前者が後者を上回っている時は、同じアーキテクチャのもとで漸進

的 (incremental) なイノベーションが進む。これをカイゼン (アーキテクチャ内イノベーション) と呼ぶことができる。対して、アーキテクチャ見直しの必要性が安定性のメリットを上回った時に起こるのがアーキテクチャ革命である。これを (radical) 革命的イノベーションということもできるだろう。アーキテクチャ (下位システム間の役割分担とインターフェース) が見直される大規模な改革が行われることになる。主要ソフトウェアの大型コンピュータからパーソナルコンピュータへの移行や、パーソナルコンピュータからクラウドコンピュータへの移行などが例としてあげられる。

柔軟なアーキテクチャのいま一つの問題点は無駄の発生である。後に原因を詳述するハードウェアの性能を無駄遣いしてもソフトウェアの使いまわしを行いたい、といった状況がその典型である。アーキテクチャを安定させようとする、どうしてもシステムのあちこちに余裕を埋め込ませなくてはならず、それが無駄になる。

従来無駄遣いしても良かった能力が突然希少なものとなることもある。たとえば、1970年代にオイルショックが世界経済を襲った時、突然、日本車はその燃費の良さゆえに脚光を浴びるようになったのはその象徴と言っていだろう。オイルショック前は乗り心地の快適性や電気機器の充実などが競争の焦点であった米国自動車業界に、突然燃費の制約が加わった。そこに、従来からエネルギーコストが高かった日本の環境で育った車が、よりインテグラル (つまりバッファを大きくとらない) アーキテクチャを採用して燃費の良さを売りに殴りこんだ。ここで大きなポイントとなったのが、インテグラルで燃費の良い製品を作るためには、調整能力の高い組織アーキテクチャが必要とされたということだろう。インテグラルな設計は高い組織の調整能力を有する自動車業界に大きな力をもたらしたのである。(藤本隆宏 et al., 2009)。

以上の分析は余裕ある資源の無駄づかいの許容が、時に希少な能力の節約につながる起こることを示している。このことを逆に言えば、システム設計上で、希少となっている能力 (ボトルネック) の部分の利用効率を最大化するのが、よい全体設計 (アーキテクチャ) であるという認識に至る。そして、そのボトルネックが変化しない間は、当該箇所の漸進的カイゼンが全体システム能力の向上につながる。

もちろん、ボトルネックが大きな革新によって解消してしまうこともある。情報システムの場合では集積回路の発明がボトルネックをハードウェアからソフトウェアに移動させた例がそれにあたる。そうなると、新しいボトルネック部分を最大効率で使えるように、システム全体のアーキテクチャが革命的に変化するようになる。ハードウェアとソフトウェアの間に OS (基本ソフト) をおいて、ハードと応用ソフトを論理的に切り離すなどはその典型例と言ってい。

このように、カイゼンとアーキテクチャ革命はどちらが一方的に優れているというものではなく、技術や市場の状況に左右されながら、企業の競争優位を作り出していくものである。アーキテクチャ革命が起ころうとしているのを、無視して保守的になることは企業の命取りになりうるし、カイゼンの局面で大きな変化を求めるのは無謀である。CIO はその動向を見極めながら、的確なシステム進化のプロセスをマネジメントしていかなければならない。

4. 情報システムのアーキテクチャ

情報システム分野における、アーキテクチャ概念の意義の高まりを理解するためには、なぜ、かつ

ては一体だったソフトウェアとハードウェアを論理的に分離したルースなアーキテクチャを採用しないといけなくなったかについての背景理解が必要だろう。

筆者はこのことを端的にシステム進化におけるハードとソフトの進化スピードのアンバランスの帰結であると理解している。すなわち、トランジスタを集積させる技術の急速な進化（いわゆるムーアの法則）によって、1960年代以来、ハードウェアの性能は級数的に向上してきた。これに対して、ソフトウェアの生産性は（さまざまプログラム言語の導入や、機械アシストによる生産技術が向上することはしたが）、ハードウェアの加速度的な進化には追いつくことはできなかった。

ここで課題となったのが、ハードウェアを更新するたびに、ソフトウェアを書き替えるのでは、ソフトウェア生産がボトルネックになって、システム全体の進化のスピードが落ちてしまう現象である。そこで、ハードウェアの性能を多少無駄遣いしても、旧世代のハードウェアのために作ったソフトウェアを使いまわす考え方がとられた。そのためにはハードウェアとソフトウェアの論理的構造化が必要となり、それを実現するために、ハードウェア—基本ソフト—応用ソフトという三層構造でシステムを組むアーキテクチャとなったのである。その後の展開を見ると、同じような論理がソフトウェア内部でも起こり、ソフトウェア内のレイヤ構造化が進んでいくことになる（加藤和彦，2016）。

このようなアーキテクチャの基本的な論理を理解すると、第3節で述べた、アーキテクチャに安定性と柔軟性のパラドクスがここでも表面化していることに気づく。一つは構造を安定化させる静的な側面である。システムの構造（アーキテクチャ）が連続的に変化してしまうと、下位システム（上記の例でいえば応用ソフトウェア）まで常に書き換えなくてはならない。そこで、応用ソフトと基本ソフトとの役割分担や、信号のやりとり（インターフェース）は安定させて、変化があまり起こらないようにする。逆にいえば、その部分が安定してさえすれば、ハードウェアの部分がどんどん進化していつても構わない。

このような安定したアーキテクチャは逆に言えばシステム全体の進化を止める効果ももっている。その矛盾が大きく表面化した時にアーキテクチャそのものが大きく変化することがある。たとえば、メインフレームの時代からパーソナルコンピュータの時代になった時には、かつて集中していた機能やデータが分散し、インターネットの登場によって分散したシステムが協調的に機能するようなアーキテクチャが生まれた。ウェブ技術の登場がその流れを強力に後押ししていった。

そして2000年代半ばからはクラウドの登場によって、いったん分散した情報が「仮想的に」クラウドに集中する現象が進行したのは記憶に新しい。これも、いったん分散した情報システムが仮想的に集中するという、大きなアーキテクチャの変化としてとらえることができる。この分野において、日本が決定的に遅れてしまっていることが、現在日本産業にとって大きな課題となっている。このことについては後節で少し踏み込んで考えてみたい。

5. 製品と組織のアーキテクチャ

アーキテクチャ概念は製品だけに限らず、それを生み出す組織構造の記述に適用可能だ。組織も人工物だからである。Simon（1981）の言うように人工物の階層化が人間の認知限界に由来するものであるならば、およそ大規模の人工物には構造化が行われる。すなわち、組織を含む全ての人工物にはアーキテクチャが存在する。経営学において、さまざまな組織構造論が展開されてきたが、それらの

議論の中でも役割分担やコミュニケーションの構造の設計などが、常に重要な位置を占めてきた。組織内の役割分担構造が中心的なテーマではあるが、近年では、コミュニケーション技術の発達によって、企業をまたがる作業を行いやすくなったこと、「複数の名刺」を持ちながら仕事をする就業形態などが一般化するなどの流れの中で仮想的な組織のマネジメントも重要なテーマとなっている。これらは全てアーキテクチャの設計と変革の問題ととらえることができる。

このように企業をめぐるさまざまなモノ（オブジェクト）をモジュールやインターフェース等の（アーキテクチャを記述する）共通の分析概念によって理解したり、設計したりすることが可能である。この作業を行うことで企業内のさまざまなアーキテクチャ間の整合性を保つ仕事の具体的な内容を明らかにしていくことができる。

6. BPR と IT サービスマネジメント

情報システムのアーキテクチャと組織のアーキテクチャの整合性を取る上で、重要な橋渡しの概念となってきたのがビジネスプロセスという考え方である。顧客視点で財やサービスが提供されるプロセスを構造的に記述し、それをより効果的に提供する方法を考える手法である。改革も含めたビジネスプロセス・リエンジニアリング（BPR）という言葉は広く一般的に使われている。

BPRが情報システムと密接に関連しているのは、情報の流れの悪さがビジネスプロセス改善のボトルネックになることが多いからと言っていいだろう。同一の顧客に別々の部門が対応していると、いったん顧客から聞いた情報を再度聞かなければいけなかったり、的を外したサービスを提供したりして、顧客を怒らせてしまったりする。社内ネットワークを整備することで、一つの窓口で同一の顧客へのサービスを一元的に提供することができるようにすることなどが、BPRの大きな狙いの一つということになる。

BPRに類似した概念としてITサービスマネジメントという考え方がある。これは、同じようなコンセプトを情報システム側で実現する手法として発達してきたものである。顧客ニーズを構造的に分析し、それを情報システムの構造としてマッピングしていく手法は、まさに経営のアーキテクチャと情報システムのアーキテクチャの整合性をとっていく作業と言っていい。同じことを技術サイドから言えば、オブジェクト指向などの情報システム構築技術が発達することによって、より顧客ニーズにそった形で情報システムを構築することが可能となってきたから実現したことと言えよう。

小尾先生が熱心に取り組みされた電子政府の分野などでも、現在サービスマネジメント的な発想での行政サービス改革の取り組みが進められている。「住民にワンストップのサービスを提供する」「いったん提出していただいたデータは二度と求めない（ワンスオンリー）」などのサービス上のゴールが設けられ、その実現に向けたシステム構築のガイドラインが策定されつつある。このような取り組みが進んだのも、小尾先生のお声かけなどで、政府CIOがアポイントされて、各府省をまたがる政府全体のアーキテクチャが意識され、統合する概念づくりが行われてきたことが大きく寄与している。

7. ビジネスモデルと技術の共進化

7.1. アーキテクチャとしてのビジネスモデル

ビジネスモデルもアーキテクチャの一種としてとらえることができるだろう。ビジネスモデルを経

営をめぐるさまざまな資源の組み合わせと、収益をあげる仕組みを構造的にとらえて設計したものととらえることができるからである。

そして、技術とビジネスモデルは共進化していく。たとえば、インターネットの登場以来、それまでは旅行代理店から買うのが通常だった航空券を航空会社から直接買う機会が増えてきた。スマートホンの普及によってその傾向はますます高まっているといえよう。これは単に技術進化の結果であるというよりも、潜在していた飛行機の予約などをより利便性高く行いたいというユーザー側のニーズが、それを可能にする技術の登場を促したという理解をすることができる。

収益モデルの変化も顕著である。かつては有償でライセンス販売されていたパーソナルコンピュータ用のソフトウェアなどを、今はグーグルなどのプラットフォーマーが無償で提供している。それに押されて既存の事業者も無償提供をしたり、月額定額制を導入したりするなど、さまざまな仕組みの工夫をこらし始めている。それに伴って、ライセンス販売時代には流行語ともなった「バージョンアップ」という言葉を最近ではほとんど聞かなくなった。無償モデルや定額制モデルの下では、新たなライセンスを売るという概念があまり意味を持たなくなってしまったからだ。

このようなビジネスモデルの変化は業界構造全体に大きな影響を与える。上記の例はいずれも、メーカーと販売代理店などのチャンネルとの関係を大きく変えている。これらの例では、中間流通の役割が低下して、生産者が直接消費者とつながる傾向が見えるが、全ての分野で同じことがあてはまるわけではないところも要注意である。

中間流通がむしろ強くなることもあるという現象をもっとも力強く示しているのがアマゾンだと言っていだろう。本稿を書いている2017年秋の時点では破竹の勢いで、PE（株価／一株あたり利益）が280を超えるという驚異的な存在になっている。

アマゾンの例は、技術とビジネスモデルを含む社会モデルがどのように共進化するか、というメカニズムを考える上で重要な手がかりを与えてくれる。単純に考えれば、ネットはメーカーと消費者を直接つなぐことが可能なはずで、中間マージンを排して直接取引が進行すると考えるのが自然である。ところが現実には、アマゾンを始めとするプラットフォーマーが間に入って取引を成立させることが多い。「技術が何を可能とするか」という視点だけでは、実際に主流になるビジネスモデルの姿は見えてこないのである。

筆者は「ボトルネックがビジネスモデルの構造（設計）を決める」という視点を導入すると、この問題が解決すると考えている。アマゾンの例で言えば、零細生産者と消費者の直接取引において欠如しているのが、相互の「信頼」である。消費者は品質や納期、決済、プライバシー保護などにおいて業者をなかなか信頼できないし、事業者側も消費者の支払いについて不安を持っている。その間をつなぎ、多様な事業者による多様な商品を一元的に扱う「アグリゲーター」の信用仲介者としての役割が大きくなっていくのである。

ボトルネックがビジネスモデルのアーキテクチャの主要決定要因であると認識すると、技術とビジネスモデルの共進化は、終わることのない連続的なプロセスであると考えることができる。システムのボトルネックが消えることはないからだ。ある技術革新が既存のボトルネックを解消して、新しいビジネスモデルのアーキテクチャを生み出した瞬間に、改善したい次のボトルネックが浮かびあがってくる。そのボトルネックが既存のボトルネックと同じものであれば、カイゼン型のイノベーション

が行われる。全く別ののであれば、(その突破を目標けた技術革新が成功した暁には)アーキテクチャそのものが置き換えられるアーキテクチャ革命がおこる。

ときにカイゼンが進み、ときには革命がおこる技術とビジネスモデルのアーキテクチャ共進化の中をどのようにナビゲートするかは企業戦略そのものであり、CIO以上にCEOの仕事なのだろうが、その中で、情報システムが戦略をしっかり支えていく必要がある。さらには、情報技術がボトルネックを解消する主要な要因となることも多く、その専門家として企業戦略全体に対してリーダーシップを果たしていく責務を担っているのが今日のCIOとっていいだろう。

7.2. ボトルネックとしてのセキュリティとプライバシー

今日のCIOにとって、考慮しなければならない大きなボトルネックとしてセキュリティとプライバシーがある。どんなにすばらしい利便性を提供するシステムであっても、この両者において問題があるものは実際には提供できない。一般に情報システムのセキュリティには機密性 (confidentiality : 秘密が漏洩しないこと)、完全性 (integrity : 情報が真正であること)、可用性 (availability : 使える状態を維持すること) の三要素があると言われているが、そのいずれもネットワークにつながれている環境の中では、常に脅威にさらされていると考えなくてはならない。プライバシーについても、守るべき顧客の重要情報を流出させたりすると、企業の信用を大きく毀損し、場合によっては存立そのものを脅かしかねない脅威となる。

これらを分けて考えるべきなのは、要件間で矛盾したりすることがあるからである。たとえば攻撃を受けている時に、データの完全性を守るためにシステムを止めてしまうことが、可用性の観点からはのぞましくない、といった事態に追い込まれることがある。何を重視すべきかは業種や守ろうとしているシステムの種類にもよるのが実態で、時にCIOは優先順位をつける難しい判断を下さなければならなくなる。

セキュリティやプライバシーを実際に守るためには、システム部門だけでなく全社的な取り組みが必要であるという視点が必要である。秘密漏洩などは、社員のうっかりミスや、基礎的セキュリティ知識の不足によるウイルス感染などによって引き起こされる場合が多い。全社員向けの日常的な啓蒙が必要であるし、さまざまな部署で構築されるシステムの導入にあたって統一されたセキュリティ基準の浸透が必要となってくる。CIOのリーダーシップが求められるゆえんである。

8. クラウドのビジネスモデルに乗り遅れた日本

スマートホン以前の携帯電話をつかったインターネット接続では世界をリードしていた日本の情報産業が、クラウドの時代になって決定的に後れをとってしまったことが、情報産業だけでなく、日本の産業全体に暗い影を落としていると言っていいたいだろう。本稿の論理にそって言えば、単なる情報システムの後れだけではなく、ビジネスモデル全体が後れ始まってきている。

焦点は、商品やサービスをめぐる情報がビッグデータとしてクラウドに集約されつつあることだ。そしてビッグデータを管理しているプラットフォーム事業者が、個別のユーザーの好みも、集合としての消費者の全体動向もよりの確に把握できるようになり、価値提供の「頭脳」の役割を果たしつつある。その中で、各デバイスはその頭脳に従って機能を果たす部品と化していく。端的に言えば、た

例えば自動車産業において、主導権を握るのがグーグルになるのか、自動車会社になるのか、という点が競争の焦点になりつつある。いち早くクラウドの基盤的ハードウェアを世界的に整備し、大量のデータを扱う経験を蓄積しているグーグルが少なくとも脅威であることは否定しがたい。一方で自動車に関する知識の蓄積という意味では、もちろんトヨタをはじめとする自動車メーカーに軍配があがる。そのどちらが強いかを固唾をのんで展開を見守っているのが現状である。

どちらが強いかという問いは、どれほど総合的な情報プラットフォームが強いのか、あるいは専門的なプラットフォームの方が強いのか、という問いに翻訳することができるのかもしれない。まさにアーキテクチャ問題で、その見極めを行うのが今日の CIO に求められている仕事である。間違えると企業を破綻に追い込みかねない重責だ。

9. IoT 時代の CIO—巻き返しに向けて—

企業戦略の一般論として、ある局面（アーキテクチャ）で決定的に出遅れてしまった時に狙うべきは、単にキャッチアップするのではなく、次の局面（アーキテクチャ）を予想して、いち早く投資をして先回りをするのである。こうすれば、現在の勝者には、イノベーターのジレンマ（Christian-
sen, 1997）といわれる、既存アーキテクチャの優位が新アーキテクチャに移る際のブレーキがかかるので、出し抜くことが可能となる。

そうすると、次のアーキテクチャが何になるのか、というのが今日の CEO, CIO が考えるべき最重要ポイントということになるだろう。予言者ならぬ筆者がそれに対する答えがあるわけではないが、変化を引き起こす可能性がある要因を二つあげることができる。

1つは「ムーアの法則の終焉」である。上にも述べたように、現在の情報システムのオープン性の高いアーキテクチャは、突出したハードウェア能力（特に演算速度）が回路の集積技術の高度化によって進んだことに起因していた。ところが、いま、集積技術が極限に達することで、ムーアの法則の終焉が予測されている。その後もハードウェアの改善は進んでいくことだろうが、そのペースが相当程度下がる可能性が高い。

仮定の問題として、ハードウェアの進化がソフトウェア生産性の改善を下回る状況を考えてみよう。この論文の論理からすると、ソフトウェアはボトルネックとなったハードウェアの性能をより効率的に利用するために、それぞれのハードウェアに合わせてつくる、クローズドなアーキテクチャに逆戻りすることも考えられる。もし、それが起こったら、オープン性を武器に巨大な力を獲得してきたグーグルなどのプラットフォーム事業者が競争力を失って、よりカスタマイズされ、ハードとソフトが一体化したシステムを提供する事業者の時代が再来するかもしれない。このあたりの見極めが経営者にとって、決定的に重要なものとなってくるだろう。

2つめは「IoT のアーキテクチャ」である。ヒトだけでなくあらゆるモノをつなぐ IoT の時代がいよいよ本格的に到来している。当初はフィールドの状態を感知するセンサーが主体であるが、自動運転などを展望すると、一步踏み込んで、機械をネットワークを介して制御する領域に突入することは必須の流れだ。

ここで考えてみたいのが、IoT のアーキテクチャが、これまでのクラウドをベースとしたものと同じとなるのか、異なるものとなるのか、という点である。この問いへの答えに大きく影響を与えると

考えられているのが、反応速度である。クラウドを使って事務作業をしたり、映画を見たりなどしている時は、1秒程度の遅延があっても大丈夫だったが、自動運転車などでは、判断の一秒の遅れは人を殺しかねない。クラウドが物理的に遠すぎることで遅延の原因などとするならば、IoTの頭脳はもっと近いところになければいけないのではないか。このシナリオがもし正しければIoTのアーキテクチャは、従来のクラウドをベースとしたものとは大きく異なるものとなり、追撃者にとってはチャンス到来ということになる。それを、いち早く察知して適切な投資を行うCIOを時代が求めていると言っているだろう。

10. おわりに

本稿を締める前に私事になるが小尾先生との長いご縁について触れておきたい。小尾先生は筆者が学生時代に熱心に取り組んだAIESEC（国際経済商学学生協会）の大先輩として、十代のころからご指導をいただいた。情熱とビジョンをもった行動力に脱帽だった。当時は東京大学にいた筆者が、その後、小尾先生ご出身の慶應義塾大学に奉職し、小尾先生は早稲田大学に行かれるなど、本拠地はすれ違い気味になったが、常に声をかけ続けてくださったご恩は忘れがたい、さらには、情報産業という共通の関心フィールドがあった。高齢化問題との関係について早期から関心を持たれて活動をされていたのは慧眼だったと敬服している。筆者はどちらかというと仕組みの方に研究の目が向いてCIOについては手薄だったが、国際CIO学会にお声がけくださり、考える機会を頂戴したことを感謝している。そこで、今回は普段考えていることをCIOの文脈で語らせていただくことにした。これも小尾先生のおかげである。小尾先生の今後もご活躍下さることを祈念したい。

参考文献

- Aken, B. R. (1989). "Large systems and Enterprise Systems Architecture." IBM Systems Journal 28(1): 4-14.
- Christensen, C. M. (1997). The innovator's dilemma: when new technologies cause great firms to fail. Boston, Mass., Boston, Mass.: Harvard Business School Press.
- Gupta, Y. P. (1991). "The chief executive officer and the chief information officer: The strategic partnership." Journal of Information Technology 6(3-4): 128-139.
- Hopper, M. D. (1990). "Rattling SABRE—New Ways to Compete on Information." Harvard Business Review 68(3): 118-125.
- McFarlan, F. W. (1984). "Information technology changes the way you compete." Harvard Business Review 62(3): 98-103.
- Simon, H. A. (1981). The sciences of the artificial. Cambridge, Mass., MIT Press.
- 藤本隆宏, et al. (2001). ビジネス・アーキテクチャ：製品・組織・プロセスの戦略的設計。東京，有斐閣，2001.4.
- 藤本隆宏, et al. (2009). 製品開発力：自動車産業の「組織能力」と「競争力」の研究。東京，ダイヤモンド社，2009.10.
- 藤本隆宏 (2017). 現場から見上げる企業戦略論：デジタル時代にも日本に勝機はある。東京，KADOKAWA，2017.7.
- 加藤和彦 (2016). IoT時代のプラットフォーム競争戦略：ネットワーク効果のレバレッジ。東京，中央経済社。