

コラム



温故知新 —すべての分野と かかわる IT, 他から学ぶことも多い—

ひろのかずお

技術評論家
hiro@khirono.com

前回、ITとその隣接分野について知の体系を作るのはどうかなどを書いてしまった。地球の歴史は45億年、原始生物が誕生して35億年、人類は500万年。「数える」のは7～8千年、計算の道具は5千年、コンピュータは半世紀余りの時が経った。知の体系の話題も含め、この辺で温故知新、歴史に学んでみる。

まずは、創造による進化を目指し思考を飛躍させるにはどうするのか。A. Einsteinは、思考の飛躍—考え翔ぶ—には、日常生活の経験・体験・大原則・理論が前提だとして、その関係を図示している。これら3本の糸から黄金の組紐を編もうとするのは、ITの世界でも同じだ。ものごとが正しく認識できれば、なにゆえかあるのかあるべきかについてより深い洞察が得られる。技術者は、与えられた課題の中に単純で美しい秩序や構造を見出し、たとえばプログラムとして結晶させる。時には新しいパラダイムを持ち込み、より複雑で高度なシステムを構築していく。

現象や課題に潜む構造を解き明かすというのなら、科学哲学上 Einstein と同じ立場（实在論, Realism）をとり、ツイスター理論や量子脳、計算不可能性を示す Penrose タイルで知られる R. Penrose が分かりやすい。彼は、日常生活の中にある現象や経験が、物理的世界・心の世界・Platonic 世界で構成されていると説く。もっと分かりやすい例として、文化人類学者 C. Lévi-Straus の「料理の3角形」がある。生もの・火にかけたもの・腐らせたもの3つを頂点とし、料理を構造化し、自然との対比で文化の進化を診る。これらからも、現実や課題に対する理解への確かな方法論のあることが伺える。

ところで、知の体系化（構造化）を進めた例に、数学者たちのグループ Bourbaki がある。それは文化のコネクターとしても評価され、C. Lévi-Straus や心理学者 J. Piaget などとの共同作業で知られる。京都賞も受賞した A. Weil がリーダ格で1935年に創設され、その成果は1939年の「集合論 要約」を手始めに、今や10部門・60以上の章・7000ページの浩瀚の書「数学原論 Éléments des mathématique」となる。記述のスタイルは公理的で、多くの場合一般から特殊へと展開し、主題の歴史的状況を示す覚書も付す。Bourbaki では、執筆者は与えられた主題を1～2年間かけて執筆し、コングレ（集会）での評価を受ける。主題や「抽象の小包」（基礎的な概念や道具立て）に対する精緻な議論の後、新たな執筆者が決められ改めて執筆が始まる。このプロセスは8～10回も行われるのが普通だったという。

このグループの最盛期は 1950 年代前後といわれるが、衰退の要因は IT と情報数学の浸透にもある。IT 技術の進歩がなければ、最適化や計算量に関する理論は生まれなかっただろうし、幾何学の復権もなかったろう。

コンピュータを用いての計算とその方法には、今や多数の研究者や技術者がかわり、IT 分野での数学的活動は広がっている。他方、どんなに有能な人が集まってもすべてを視野に収め、全体を把握し、統一的に論ずることは、IT の場合もそうだが、数学の世界でもできなくなってしまうらしい。科学史家 M. Demazure は、「Bourbaki 流の数学ではプロセスのすべてを描ききれない。アルゴリズムまたはプログラム関連の事柄のすべてを包括する枠組みを見出すことは明らかに不可能だ」と指摘する。数学の情景が変わったと表現する人もいるが、数学と IT との相互作用の深さと広さにも驚いてしまう。

ある推定によれば、数学の世界では毎年 20 万以上の定理が発表されているという。ちなみに、数理モデルを専門にする J. L. Casti は、20 世紀を動かした 5 大定理ということで、J. von Neuman のミニマックス定理、L. E. J. Brouwer の不動点定理、H. M. Morse の変分法、A. Turing の停止定理、G. B. Dantzig の単体法を挙げている。彼の選定基準は、重要性、美しさと適用範囲の広さ、応用性、証明方法、哲学的示唆だ。これを参考に、移ろいやすい IT の世界だが、温故知新で何かを選び学ぶという場合の基準を作ろう。

- (1) IT の進化過程で重大な障害を取り除いたか。この結果、IT 探求の新分野を拓いたか。
- (2) 単純で美しいか。大量の知識を簡潔に整理しているか。IT の世界の道標となるか。
- (3) 重要な応用例がありそうか。自然界や人間社会をより豊かでやさしさに満ちたものにするための基礎になっているか。
- (4) 新しい技法や考え方を含むか。それらは他の重要な問題にも利用可能か。
- (5) 人間や社会についての未知の重要な事実を教えてくれるか。宇宙、世界、人間自身に対してさらに深い洞察を与えてくれるか。

この基準で選んだ 1 冊に、Turing 賞受賞論文集「ACM Turing Award Lectures, The First Twenty Years, 1966 ~ 85」がある。本書では、研究や技術開発の動機となったアイデア、技術的な構想が実現されていく状況や背景、技術的な役割や貢献などが示され、今でも有用な示唆に富む。

プログラミングの現場は、今やメインフレーム時代と

すっかり変わってしまったが、試みに、プログラミング言語周辺で論文のつまみ食いをするのも悪くない。

たとえば、Web 上のコンテンツ言語 Curl や、Java での拡張努力の成果の 1 つ JCSP (Java の CSP Class ライブラリ)、さらには RDF (データ構造記述言語) ベースのプログラミング言語として Fabl とこれに対抗して生まれた Adenine など、言語の話題には事欠かないが、これらの行方を見るのにも役立つかもしれない。

— CS (Computer Science) は、計算理論・プログラミング言語・教育の 3 つの領域において、「形と中身の混同」と名づけるものを出発点としている (フレーム理論の M. Minsky)。問題解決におけるプログラミング言語は、計算のほとんどを記述するための手段であり、思考に影響を及ぼす表記法としての役割と、抽象計算機械に対する指令としての役割を果たしている。表記法という点では、表記方式の持つ暗示性や経済性というような一般属性を含む特性が問題解決に有用だ (APL の発明者 K. E. Iverson)。上の役割を果たし、プログラミングのための美しい道具と、良いスタイルを助長する言語を作ることが必要だ。なぜならプログラムは優美さとスタイルを持つことができるし、持つべきだからだ (フォント設計アルゴリズムや $\text{\TeX}$ の D. E. Knuth)。強力でエレガントな上に、数学的にしっかりした意味を持つプログラミングスタイルもよい (FORTRAN や言語記述の J. W. Backus)。ともあれ、言語やシステムの設計では、単純なことが良い結果を生む (Communicating Sequential Processes, CSP の C. A. R. Hoare)。また、直接手を下した経験とよく吟味された道具としての価値の重要性は言うに及ばない (Pascal の発明者 N. Wirth)。実際のプログラミングの面で経済性を重視するのであれば、誤りのないプログラムこそが重要だ。これは、知的に管理できるプログラムだけを作れば、技術的に実現可能である (GOTO 文批判や構造的プログラミングの E. Dijkstra)。ところで、多様なプログラミング・パラダイムの使用経験を通じて特に重要と感じるのは、初学者にパラダイムを教え、パラダイムの表現を支援するためのプログラミング言語が必要なことだ (プログラムの自動証明の R. Floyd)。……

IT は、日常生活のみならず未解決の分野—宇宙の秘密・生命の起源・人類の将来—とも深いかわりを持つ。IT にも創造による進化は望ましい。が、豊潤であらゆる方向に向いている IT 世界だ。良い先生や洗練された文献・書物からの知的収奪に積極的になり、先人の成果を大いに活用するという応用による進化も悪くはない。

(平成 15 年 6 月 16 日受付)