

未来社会を支える仕組みを実現するための研究開発

Research and Development to Achieve a System That Supports the Future Society

羽 田 昭 裕

要 約 自動化と自律化を、描いた未来社会像を実現するための核と捉えて、取り組む必要のあるコンピューティング技術について述べる。最初のコンピュータであるENIACで実現されたのは、数学者の思考の自動化であった。これを出発点として、産業革命から始まった機械の自動化は洗練され、自律化し、貨幣に関わる処理、組織活動などが自動化された。現在は、管理された組織や貨幣に関する処理の自律化、それ以外の人工物の自動化が進んでいる。

自動化と自律化の歴史を踏まえると、近い将来に必要となるのは、自動化し、自律化した人工物に対してより自律性を求められる人々を助けるコンピューティング技術であり、思考の自動化はデザイン思考やシステム思考の結果が対象となろう。この課題を考慮すると、人間、人工物、サービス、環境などを記述する道具が重要となる。一つの事例と二つの技術（メレオロジー、形式概念分析）を取り上げて、道具のありかたを示した。

Abstract We describe the computing technologies that need to be addressed on the ground that automation and autonomy is a core feature for creating ways to improve our future. ENIAC, the first electronic computer, realized the automation of the thinking of the mathematician. Let this be a starting point, machinery automation that began from industrial revolution was sophisticated and proceeded to autonomous, and a host of processes relating to money and human organization have been automated. Currently, automated processes for money and well-managed organizations are on the road to autonomy, and automation of the other artifacts is progressing.

In light of the history of automation and autonomy, new computing technology that helps those who are required to be more autonomous to correspond to the penetration of automated and autonomous artifacts will be needed in the near future. It creates demand for automation of content of thought created by using system-thinking or design-thinking. As we address the issues listed above, new practical tools to articulate human beings, artifacts, services and environment are required. One case study and two techniques (mereology and formal concept analysis) showed the way of such tools.

1. はじめに

これまでも技術の将来についての見通しは、いろいろな形で示されてきた。

1984年にアラン・ケイ^[1]は、「音楽は見えないものに形を与える」というダ・ヴィンチの言葉を引きながら、「楽器、音楽、総譜」をメタファーとして、「コンピュータ、コンピューティング、ソフトウェア」の関係を示し、コンピューティング、ソフトウェア、コンピュータリテラシーなどのあり方について論じた。そして、バッハやベートーベンのような音楽家に先行して、複雑さをなくし、単純さを実現する「オッカムの剃刀」が果たした役割の重要性を述べ、コンピュータと人間を架橋するために普及が望まれる技術として、部分と全体の関係を規定するオブジェクト指向やエージェント、その応用であるユーザーインタフェースなどを取り上げ

ている。

コンピューティングを、コンピュータの使用による実世界と仮想の往還という観点で見ると、「デジタルテクノロジー企業が創る未来社会」^[2]で描いた像を実現するための核となる技術は、自動化 (automation)、自律化 (autonomy) 技術であると捉えている。

以下の章では、自動化と自律化のこれまでの流れと今後の流れについて、自動化と自律化の関係 (2章)、出発点として数学者の思考をコンピュータで自動化した ENIAC の改造 (3章)、コンピュータでの自動化の流れ (4章)、分散並列化によってソフトウェアによる自動化に移行した経緯 (5章) を紹介し、この流れを踏まえると近い将来にはコンピューティングによる自動化に移行するであろうという見通しを述べ (6章)、そのために必要な道具の要件 (7章) を議論し、将来像を先取りしていると捉えている事例を通じて検討する (8章)。

2. 自動化と自律化

自動化 (automation)、自律化 (autonomy) についてはさまざまな定義がある。自動運転車の自律的な運転を例にして考えると、自律化とは、他の指示に因らず、周りの状況を評価することによって最適な選択肢を判断したうえで、自動化された項目をルールに従い機械が実行していくことであり、自律化は、自動化を前提としている。また、自動運転車に車単独の「自律型自動運転」および人間と車の「協調型自動運転」があるように、自律化は人間の能力の拡張と連続している。

2014年に、IEEE Computer が“2022年に開花するであろう23のテクノロジー”^[3]として公表した領域は、自動化と自律化に関するコンピュータとコンピューティングの技術を含んでいる (表1)。なお、ここに列挙されている技術については、これまでの技報および近刊の技報でおおよそ網羅している。

表1 2022年に開花するであろう23のテクノロジー^[3]

分野横断的セキュリティ問題	3D 集積回路	クラウドコンピューティング	機械学習と知的システム
オープン知財運動	マルチコア	IoT	コンピュータビジョンとパターン認識
持続可能性	光工学	ネットワーク利用と相互接続性	生命科学
MOOCs	ユニバーサルメモリ	NUI	計算生物学と生命情報学
量子コンピュータ	SDN	3D プリンティング	医療用ロボティクス
デバイスとナノテクノロジー	高性能計算	ビッグデータと解析	

自動化を自律化につなげる機械的な判断、およびそれに必要な計算能力の見通しについては、2006年の技報論文^[4]でデータ解析と高速処理の基盤について、1952年のアメリカ大統領選挙で、CBS テレビが、世界初のプログラム内蔵方式商用コンピュータ UNIVAC-I を使って予測を行い、開票わずかの段階で当選結果を正しく予測したことにより、コンピュータによる

データ解析の威力が認知されたことに遡って概観した。

そして、人工物の自律化を補完する人間の能力の拡張については、「拡張現実感」、「ゾーニングからの脱却」、「社会的現実感」があること、そして、その表現形式（モダリティ）が言語メディアから、感覚入力を人工的な信号で置き換える技術である感覚的現実感（perceptual reality）を基礎にして、マルチ・モーダルを軸とした臨場感あるインタフェース技術に移るといふ展望を2003年の拙論^[5]で紹介した。この技術は、その後、ウェアラブルコンピューティング、BMI（Brain Machine Interface）などのかたちで実現が進んでいる。

以下では、このような機械的な判断や人間能力の拡張とならぶ、自律化の基礎である自動化について述べる。

3. ENIAC 1948 と数学者の思考の自動化

コンピュータを利用した自動化の歴史は、エッカート博士（J. Presper Eckert）とモークリー博士（John Mauchly）が設計し、ペンシルバニア大学のムーアスクール（Moore School of Electrical Engineering）で1943-45年に作り上げ、1946年2月14日に公開したENIAC（The Electronic Numerical Integrator and Computer）に遡る。ENIACは、当時のリレー計算機とは異なり、数値計算を自動化することで、「数週間かかっていた計算を15秒で解いた」^[6]と伝えられた。

ENIACは、世界最初の電子計算機であり、コンピュータ産業という枠組みを作り上げたけれども、コンピュータそのものは外部プログラミング方式であり、その後のコンピュータの原型とはならなかった^[7]といわれている。しかし、IEEE Annals of the History of Computingに掲載された連続した論考^{*1}では、コンピュータ自体よりも、のちのソフトウェアにつながるコードに注目して、フォン・ノイマン（John von Neumann）のチームが改造したENIAC（以下、ENIAC1948）が現代的なコード規範（The modern code paradigm）、すなわちプログラム内蔵方式（Stored Program Concept）の原型であった^[8]と述べている。この論考を基に、自動化の経緯を振り返る。

現在のコンピュータは、フォン・ノイマン型といわれる。ノイマン型の由来となる“EDVAC設計の第1稿（The First Draft Report of the EDVAC Design）”には三つの要素が含まれている。そのうち、コンピュータとしての二つの機構は消えていったが、コンピューティングの方式であるプログラム内蔵方式は、今に至るまで影響を残している^{*2}。プログラム内蔵方式は、なぜ、現代的なコード規範と位置づけられたのだろうか。ENIAC1948が行った最初のコンピューティングは、熱核連鎖反応すなわち水素爆弾の研究のためのシミュレーションであった。ノイマンがNew York Times誌^[11]で「ロボットとしてのENIACは、改造によって、数学的思考において人間の脳を近似した」と述べているように、ノイマンのチームは、モンテカルロ法を実現するに当たり、数学者の思考を表現するために、流れ図を考案した。そして、以下のような点で、現代的なコード規範を生み出した^[10]。モンテカルロシミュレーションの核は、乱数の発生であり、複数の手続きで共通して呼び出されるサブルーチンが組み込まれた。また、中性子の振る舞いによった計算などにおいて、状態に応じて計算の流れを変える必要があるため条件分岐が生み出された。このほか、中性子と原子核が相互作用する確率（衝突断面積）を、中性子の速度と物質の性質を表にすることで、内挿によって多項式近似できるというアイデアを実現するためにテーブルルックアップが考案された。コードの観点からすると、サブルー

チン、条件分岐、テーブルルックアップは現代的なパラダイムの始まりといえ、ENIAC1948は内蔵プログラム方式を実現する現代的な規範でのコードが動いた最初のコンピュータであるとしている。

自動化の観点から見ると、使用者であるノイマンらが、数学的思考という点から人間の脳の処理を自動化しようとしたことが出発点であり、そのために自動化の対象を流れ図として表現し、流れ図として記述された計算の流れを実現するものとして内蔵プログラム方式が求められた、という点が重要である。

4. コンピュータによる数学的思考の自動化

次に、バームが1955年にプログラマになって以来の経験に基づいて述べた2006年の論考^[12]を参考に、コンピュータを利用した自動化と自律化の流れをたどる(図1)。

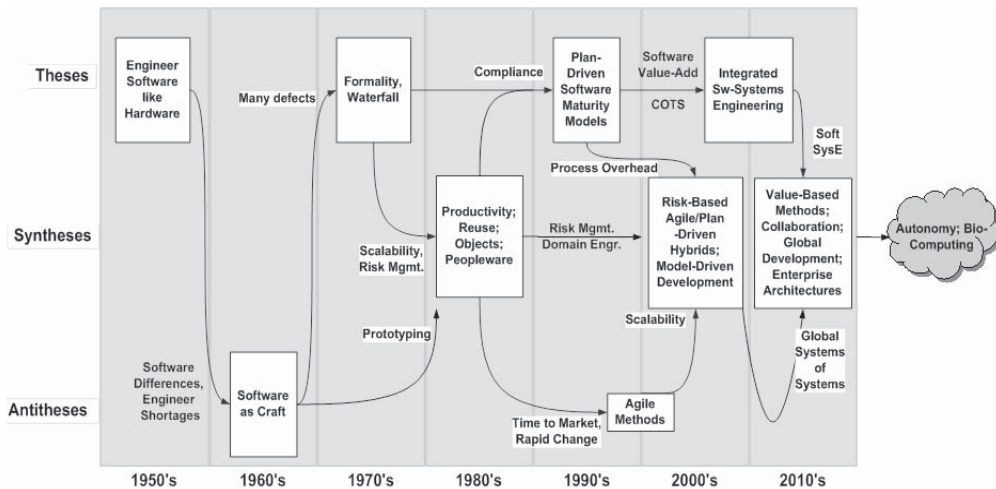


図1 ヘーゲル的なソフトウェア工学発展観^[13]

1950年代にフォン・ノイマン型コンピュータが普及し始め、ハードウェアと異なりいつでも動作が変更可能で、劣化しない「ソフトウェア」という概念が生まれた。その基本は、何か問題があって、それを解決する思考の流れをコードとして実現する、ということであった。

フォン・ノイマン型コンピュータでも、限られた領域やテーブルを、最適に分割し、割り当てる仕事やアドレスの管理などは利用者に残されていた。この作業がコンパイラによって自動化された後に、プログラムという言葉が利用されるようになった。最初のコンパイラは、グレース・ホッパー (Grace Murray Hopper) が開発したUNIVACのコンパイラ A-0 Systemであり、現在のプログラミング言語のように自然言語に近い記述ができるコンパイラは、同じくホッパーが開発したFLOW-MATICから始まった。

この後、1960年代に本格的なプログラミング言語ができて、ハードウェアを理解していない人でもプログラムを作ることができるようになったことでソフトウェアは増え始めたけれども、その副作用として、いわゆるスパゲッティコードが多く出てきた。システム開発プロジェクトの安定性と先見性に対するニーズに応えるために、問題とプログラムの間に、要求と設計を置いて、形式的に進めることが始まり、「ソフトウェア工学」という旗印のもとで1970年代

に普及した。

そして、プログラミング言語の普及によって、数学的な思考の自動化の道具としてのコンピュータの利用が進み、数学的に判断を記述できる分野での自律化、数学的に処理を記述できる分野での自動化が進んだ。後者には、事務処理と呼ばれた、管理された組織の活動や貨幣に関する処理が含まれる。

5. 分散、並列、ネットワーク化とソフトウェアによる自動化

1980年代に入ると、並列した計算や、複数のソフトウェアの連携が進み、相互運用性が高まることに伴って、コンピュータの仮想化が進んだ。このことと、ソフトウェア開発の生産性や拡張性をより上げるために、個々のプロジェクト（マイクロプロセス）をまたがってプロセスを反復できる組織力（マクロプロセス）という考えが広まったことで、焦点はコンピュータからソフトウェアに移った。1990年代にはネットワーク化、オープン化、ダウンサイジングの進展とともに、分散されたものの一貫性や整合性が課題となり、オブジェクト指向が再び注目を集めた。

そして、組織の間の垣根が低くなると同時にグローバル化が進み、問題は複雑かつ多様で、想定しきれないものが主流となり、また環境の変化も速くなってきた。マクロプロセスの重視と問題の性質の変化を背景にして問題とプログラムが直接に結びつく流れが2000年代に再び生まれてきた。例えば、汎用的な設計とモデルで個々の既存システムを統合するという、EA（Enterprise Architecture）、SOA（Service-Oriented Architecture）やSaaS（Software as a Service）などのアプローチや、要求とプログラムが直結することを目指したアジャイル開発法が普及した。プロトタイピングと実用化がつながっていくことは、積極的な表現をすれば進化するシステム、別の言い方をすれば後付けとなる。

このように分散、並列化が進み、システムの複合化（massive system of systems）やグローバル化が進み、立ち向かう問題が大きく、複雑に、そして変化の速いものとなった。そして、並列、分散、ネットワーク化することで判断が自動化できる人工物の自律化が進んだ。

6. 自動化、自律化の近い将来の見通し

前の章までで概観したように、ノイマンの流れ図からEAやSOA、そして自律化へと移るにしたがい、コンピュータによる当面する個々の問題の解決から、分散し並列した処理を一貫して動かすソフトウェアで価値を生み出すことへ移ってきた。現在は、目的の異なる自律したシステムの集まりを複合して扱うコンピューティングに移ってきている（図2）。

より長い時間軸で見ると、産業革命から始まった自動化は、機械化に始まり、科学的管理法の発達で企業などの組織活動、貨幣に関わる処理の自動化が加わった。また、コンピュータの発達により、この流れは加速され、いくつかの機械は自動化から、自律化に向かった。

現在は、管理された組織や貨幣に関する処理の自律化、それ以外の人工物（国家、法律、教育）の自動化が進んでいる。センサーデータやオープンデータなどの機械可読な（machine readable）データ、SDN（Software Defined Network）、3Dプリンターのように、データ、ネットワークやモノを、ソフトウェア的に扱えるという技術が普及したことも、この自動化、自律化の背景となっている。ベームの見通しでは、今後の流れを生み出すインパクトのある技術として自律化とバイオコンピューティングを挙げている。自律化に含まれるものとして、知的

エージェントの協調，状況に適合する自律的ソフトウェア，機械学習技術，以上のような自律化技術を応用した通常からナノスケールまでのロボットの拡張を挙げている．そして，多くの機械が自律化し，そのような機械によって組織も自動化や自律化が進行すると，焦点はソフトウェアからコンピューティングに移り，コンピューティング自体も，単にコンピュータの使用にとどまらず，ソフトウェアを介して機械や組織などの人工物の使用まで広がる概念となる．

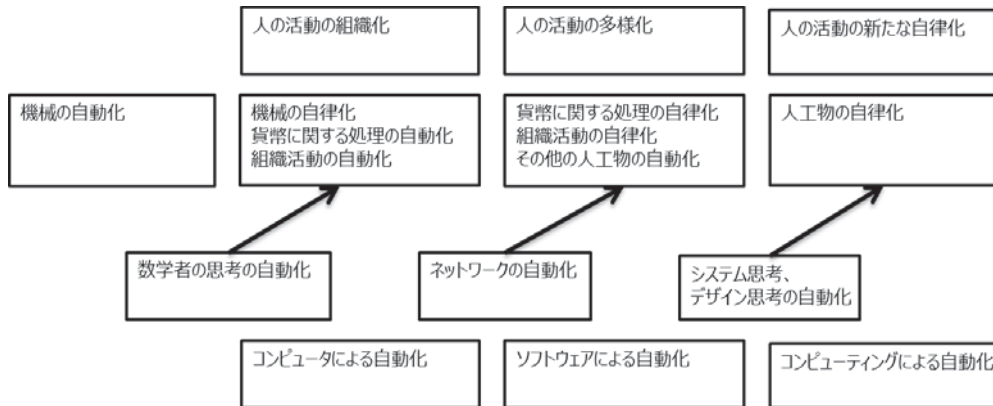


図2 自動化、自律化の流れ

では，このように進む人工物の自律化に対応する，コンピューティングによる自動化とはどのようにして実現するのがよいだろうか。

ひとつ方向性は，複雑で大きな問題，管理しきれない問題を扱うアプローチとして生み出されたシステム思考やデザイン思考の取り組みに見出せるだろう。

ここでは，システム思考をシステムダイナミクスの流れで捉える．すなわち，理解し，対応しようとしている対象をシステムと捉え，システムは何らかの目的を果たす要素の組み合わせであり，レジリエンス・ヒエラルキー・自己組織化があるため全体は要素の総和以上となり，完全には予測できないし，コントロールできないという理解に立つ．そして，システムの中のレバレッジポイントに注目し，そこへ効果的に働きかける手段を見つけ出す．これをアラン・アトキンソンはイノベーションと呼んでいる。

デザイン思考は，定まった定義はないため，ここでは存在する課題を解くのではなく，課題そのものを見つけることから始めるというアプローチはシステム思考と共有しつつ，「共感」と「体験」を繰り返して創造的に問題に取り組むことで，個々人の受容とその基礎となるメンタルモデルを考慮に入れた方法と捉える。

7. コンピューティングによる自動化のための道具

システム思考やデザイン思考に対応する道具について，オープンデータやオープンガバメントを例に考える．機械判読可能で，アクセスしやすく，無償で再利用できるという自動化の流れが出てきて，それが行政を透明化することで，市民生活の向上，技術革新，経済成長，雇用創出を目指す新たな自律性の流れと，相互に影響を与えている．一方で，この二つの流れの間にはギャップが存在しているのが現状である．例えば，機械判読可能なデータは，人間が直接読み取ることが目指していない．これは，外部プログラム方式のENIACと，数学的思考に基

づいた自然で柔軟な記述を求めるノイマンらロスアラモス研究所との対比に似たギャップと見える。このギャップを埋めるには、新たな自律性に対応する、ノイマンの流れ図に相当する記述法が求められる。それは、数学者の思考ではなく、実世界についての構想を記述するものである。関連するものとして、オープンデータ、オープンガバメントでは、米国政府でのセキュリティ分野を中心としたNIEM (National Information Exchange Model) や米国国防省のUcore (DoD Universal Core)、欧州ISA joinup のコア 語彙などが挙げられる。これらの取り組みは行政サイドの相互運用を中心としている。

この問題は、ダグラス・ロス (Douglas T. Ross) に遡ることができる。ロスは、設計、問題解決、プログラミング、ましてや数学から出発するのではなく、世界観 (world view) を具体的に、実行可能なものとできれば、問題解決プロセスや、問題解決チームにおける未知の人間の要素に取り組む基礎となると述べている。世界観とは、人間と世界にかかわることがらを全体として意味づける見方である。そのアプローチを基礎にして、モデルの記述から実物の設計図を作り出すCAD (Computer-Aided Design) や、情報システムの設計法 (SADT: Structured Analysis and Design Technique) などを生み出した。現在、開発環境といわれているものの源流であるCASE (Computer-Aided Software Engineering) は、CADのアプローチを情報システムに適用したもので、そのためのモデル記述のしかたはSADTやIDEF0に由来するといえる。

より動的で実行可能な世界観のさきがけの一つである、青山学院大学がスタンフォード大学と共同して1997年に開始したプロジェクト^[14]は、当時の市場の変化を鳥瞰して仮想環境上に部分的に実現し、その環境で新製品を企画し、その製品を仮想企業で開発する実践的な教育を目指していた。そして、新製品企画 (市場と商品構造の理解・分析、意匠設計)、事業計画、マーケティング・リサーチ、エンジニアリング (製品の基本設計) が、多層に並列して進み、自律協調して競争する世界を仮想的に実現した。ただし、この時点ではシミュレーションにとどまっており、現実との往還は共同演習に限定されていた。

現実との往還を含む、実世界についての動的で実行可能な構想の記述について考察するための事例として「大規模ゲノム疫学研究の統合情報基盤」^[15]を取り上げる。

8. コンピューティングによる自動化の事例

この事例は、以下に説明するように、個別化医療や先制医療を現実のものとするために慢性疾患の研究に求められる横断的なアプローチを実現するという世界観に基づくコンピューティングであり、研究者の行動に注目して研究プロトコルからメタデータを自動生成することで効率化するプロトタイピングを実現している。そこでは、大型で分野横断型の新たな試みに求められる共感とプロトタイピングに加えて、倫理的な視点を含む綿密な事業計画の策定と、組織構築や事業の円滑かつ効率的な実施を可能とする制度設計、そしてそれを支えるコンピュータとなる情報基盤といった実世界の構想の記述に基づいた事前の用意が大切となる。

個別化医療については、従来からの予防と合わせることで健康長寿社会を作ろうという考えが普及している一方、実現が難しいことも共有されつつある。難しさのひとつは、主に予防しなければならないのは慢性疾患であるため、生まれてから今までの行動と経験の積み重ねや体質の多様性が影響するという慢性疾患の特性に対応した縦断的アプローチの難しさであり、この解決のために長期にわたるゲノムコホート研究が始まっている。この事例を通じて、複雑な

問題の解決に、コンピューティングをどう使うかを概観する。

縦断的なアプローチは、法令やガイドラインを遵守しながら、食生活や生活習慣を追跡し、ゲノム・蛋白質や代謝物などの網羅的解析、それらが健康や発病にどう結び付いたか調査する、という行動に分解できる。そして、研究者が定義する「研究プロトコル」(研究計画)に基づいて行動が定義され、発見や推論の基礎となる。注目するのは、研究者の自律性である。研究が継続されない限り、最新の知識と最良の技術があっても、今治せる病気しか将来も治せない、という事実にもかかわらず、肝心の研究者が何をしていて、どんな課題を持っているのかについては、あまり注目されていない。そこで、この研究プロトコルから、メタデータを自動的に抽出できれば、それがレバレッジとなり、「縦断的なアプローチ」が促進されるだろう、というアイデアである。データではなく、メタデータであるのは、扱うデータが個人データなどセンシティブな情報やサンプルであることに加え、集めるデータはその時点で存在せず追加的に調査や集計をするケースもあり、全体的な方針を定義した時点では対応する個別の要素が確定できないためである。

これをプロトタイピングするには、プライバシー保護とデータ活用の両立、組織や企業を横断した社会的・政治的な調整、データの統合・数理統計・言語や画像処理などヒト生命情報の活用に必要な技術基盤といった実現が難しい課題が含まれている。これらを解決する、あるいは懸念を解消しないと、プロトタイピングに納得性が出ず、共感や協力が得られない。そのためには、技術を用意し、組み合わせ、仕組みを作ることに努力と時間をかけた取り組みが求められ、情報基盤は3年かけてプロトタイピングできる状況になった。

このような仕組みの構成と記述には、人間と世界(人工物、サービス、環境)についての意味を、データの由来と系統についての情報などを用いて関連するデータをまとめ上げ、データの利用をガイドする道具が役立つ。そのようなコンピューティング技術の候補に、形式概念分析とメレオロジーがある。形式概念分析(FCA)^{*3}は、思考の単位を概念と捉え、その概念の束を構造化することで、論理的な思考に基づくコミュニケーションに貢献することを目指すデータ分析の方法論である。メレオロジー^{*4}は、構造化された対象領域で、要素への完全な還元を前提にできない対象についての計算をモデル化できる。

9. おわりに

以上のように、自動化/自律化の焦点は、マイクロプロセス/マクロプロセス、部分/全体の対比などと関連しながら、コンピュータからソフトウェア、そしてコンピューティングに変わっている。冒頭の「楽器、音楽、総譜」のメタファーを用いると、焦点は楽器から総譜、そして音楽に移ってきた。西洋音楽が18-19世紀に、演奏自体が作品であった状態から、聴く人を動かすことが焦点になり、演奏者や指揮者への正確な総譜を作品として書く時代になったように、コンピューティングという音楽は、自律した機械や組織と、それに向かい合う人々を動かす構想の全体像を総譜で表現する作曲家=構成者によって書かれ、演じられると想定している。

作曲家=構成者として新たな自律性を求められる人々には、コンピュータ志向の自動化や自律化との狭間を埋め、人工物と人間とがともに理解できる記述法とその実行環境が求められる。その総譜となる思考を記述する道具として、実世界と仮想の往還を支える技術が、実行環境として揃いつつある。

このようなコンピューティング技術は、構成者としての新たな自律性を情報技術で支えること、情報技術を自動化・自律化の方向で進化させること、この双方の基礎となると考える。

- * 1 ENIAC の歴史をめぐる、連続した三つの論考の二つ目。一つ目^[9]は、プログラム内蔵方式の歴史と代替案についての考察。二つ目のこの論考は、プログラム内蔵方式 (Stored-Program) の実装という観点から ENIAC の歴史について考察^[8]。三つ目は、コンバージョン後の最初に走らせた、モンテカルロシミュレーションのプログラムとしての詳細^[10]。
- * 2 “EDVAC の三つの側面のパラダイムとしての影響力は逸れていった。まず、トランジスタとコアメモリが、真空管と遅延線を時代遅れにするにつれて、ハードウェアパラダイムとしての適合性は消えていった。フォン・ノイマンのアーキテクチャ設計はより長生きしたが、そのミニマリズムは、並列処理、MPI、命令パイプライン、DMA、スタック、アドレス可能レジスタなどの発明によって消し去られた。対照的に、現代的なコードパラダイムは少なくともプロセッサによって実行される機械語の記述としては一ほとんどそのまま現存している。”^[9]
- * 3 ここで「形式」とは、数学的基礎を持つことを意味する。FCA における概念はデータとして扱われ、特定のオブジェクトの集合 (外延) と特定の属性の集合 (内包) の間の関係 (コンテキスト) で表現される。FCA は、多くの領域で現れる通常の間人活動を、そのようなデータとして扱い、入力データから 2 種類の出力を導く。一つ目は概念の束 (concept lattice) である。概念の束は、下位概念-上位概念関係によって階層的に順序付けられたデータにおける形式概念の集まりである。FCA は、「データの中からの概念の発見と推論」、「データの中からの依存性の発見と推論」、および「可能性を折り畳んだり/拡げたりすることでのデータ、概念、および依存性の可視化」という、データと知識の概念処理の三つの要素を自然に統合することの特徴としている。使用者が概念の束を見ながら行うことで、このような概念処理を支援するツールはいくつか存在し、多様な問題に適用されている。例としては、よくある話題に基づく概念への web 検索結果の階層的組織、遺伝子発現データ分析、情報検索、ソフトウェア工学におけるソフトウェアコード、デバッグ、データマイニング、および設計の分析と理解、ドキュメントと e-メール収集の分析と組織を含むインターネットアプリケーション、注釈付きの分類法、などの多様なデータ分析プロジェクトを含んでいる。(文献^[16]による)
- * 4 メレオロジーは、部分を意味する希臘語「メロス」(meros; μέρος) から来ており、集合論に基づかない、部分と全体に関する一階の述語論理体系である。メレオロジーは、部分が集まって全体ができ、全体を分割すると部分が得られるというのが基本原理であり、諸部分と諸部分の和によって成る融合体を新たな諸部分と認めていることが特徴である。このことで、与えられた複数の対象を合わせて一つの対象とみなしてしまう。また、全体から部分への分割は、概念図式を用いておこない、異なる概念図式を用いれば、異なる分割となる。
メレオロジーの応用として D. ビョルナー (Dines Bjørner) のドメイン科学 (Domain Science) がある。ビョルナーは、ソフトウェアに先立つ要求は、ドメインに規定されるものとみろという立場をとった上で、ロスの考えを受け継ぐものとして、人間と要素とするドメイン、あるいは人工物としてのドメインについての科学的な方法^[17]を打ちたてようとしている。それをドメイン科学と呼び、ドメイン科学とドメイン記述 (Domain description) をつなぐものをドメイン工学 (Domain Engineering) と呼んでいる。そして、ドメイン科学は、メレオロジーと関連が深く、いくつかのシステムについてメレオロジーはラムダ式に対応することを示している^[18]。

参考文献 [1] Alan Kay, “Computer Software”, Scientific American, September, 1984, Vol.251, No.3
 [2] 小椋則樹, 「デジタルテクノロジー企業が創る未来社会」, ユニシス技報, 日本ユニシス, Vol.35 No.2 通巻 125 号, 2015 年 9 月
 [3] Hasan Alkhatib, et al., “IEEE CS 2022 Report”, IEEE Computer, 2014. <http://www.computer.org/web/computingnow/2022-Report> (2015 年 7 月 29 日確認)
 [4] 宮地恵美, 「寄稿 技報 90 号の発刊によせて」, ユニシス技報, 日本ユニシス, Vol.26 No.2 通巻 90 号, 2006 年 8 月
 [5] 羽田昭裕, 「E テクノロジによる現実と仮想の架橋」, ユニシス技報, 日本ユニシス, Vol.22 No.4 通巻 76 号, 2003 年 2 月
 [6] T. R. Kennedy, “Electronic Computer Flashes Answers, May Speed Engineering,” The New York Times, February 15, 1946.
 [7] 高橋伸夫, 「殻」, ミネルヴァ書房, 2013

- [8] Thomas Haigh, Mark Priestley, Crispin Rope, “Engineering “Miracle of the ENIAC”: Implementing the Modern Code Paradigm”, IEEE Annals of the History of Computing, Vol.36 No.2, April-June 2014.
- [9] Thomas Haigh, Mark Priestley, Crispin Rope, “Reconsidering the Stored-Program Concept”, IEEE Annals of the History of Computing, Vol.36 No.1, January-March 2014.
- [10] Thomas Haigh, Mark Priestley, Crispin Rope, “Los Alamos Bets on ENIAC: Nuclear Monte Carlo Simulations, 1947-1948”, IEEE Annals of the History of Computing, Vol.36 No.3, July-September 2014.
- [11] Will Lissner, “‘brain’ speeded up for war problems”, THE NEW YORK TIMES, December 13, 1947.
- [12] Barry Boehm, “A View of 20th and 21st Century Software engineering”, ICSE '06 Proceedings of the 28th international conference on Software engineering, 2006.
- [13] Barry Boehm, A Hegelian View of Software Engineering Evolution, “A View of 20th and 21st Century Software Engineering”, ICSE 2006 Keynote Address, May 25, 2006.
- [14] 羽田昭裕, 「青山学院バーチャル・キャンパス基盤システムの構築」, ユニシス技報, 日本ユニシス, Vol.20 No.1 通巻 65 号, 2000 年 5 月
- [15] 沖俊吾, 「大規模ゲノム疫学研究の統合情報基盤の構築事例」, ユニシス技報, 日本ユニシス, Vol.34 No.3 通巻 122 号, 2014 年 12 月
- [16] Radim Belohlavek, “Introduction to Formal Concept Analysis”, Dept. Computer Science, Palacký University, 2008
- [17] Dines Bjørner, “Domain Science & Engineering: A Foundation for Computation for Humanity”, Computation for Humanity: Information Technology to Advance Society, CRC-Press, 2013
- [18] Dines Bjørner, “A Role for Mereology in Domain Science and Engineering”, Mereology and the Sciences Parts and Wholes in the Contemporary Scientific Context, Springer International Publishing, 2014

執筆者紹介 羽田 昭 裕 (Akihiro Hada)

1984 年, 一橋大学社会理論課程卒業. 爾来日本ユニバック(株), 日本ユニシス(株)勤務.

意思決定支援ソフトウェアの開発・適用に従事. その後, 業務システムとその基盤の要求分析・開発, エンタープライズ・アーキテクチャや情報活用など企業の IT 活用力向上を支援するコンサルティングを経て, 2007 年総合技術研究所先端技術部長. 2011 年より同研究所長.

情報処理学会会員.

