

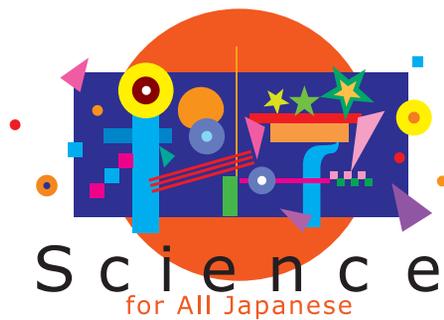


21世紀の科学技術リテラシー像～豊かに生きるための智～プロジェクト

情報学

専門部会報告書

平成20年(2008年)6月



科学技術の智プロジェクト
研究代表者 北原和夫（国際基督教大学教養学部）
<http://www.science-for-all.jp/>

「科学技術の智」プロジェクト専門部会報告書の刊行にあたって

「科学技術の智」プロジェクト委員長

北原和夫

全ての日本人が身に付けてほしい科学技術の基礎的素養を明示しようというプロジェクトを開始したのは、2005年であった。若者の理科離れが進んでいる現状にあって、2003年に日本学術会議は「若者の理科離れ特別委員会」（後に「若者の科学力増進特別委員会」と改称）を組織し、その現状打開のために何をなすべきかについて検討を始めた。その結果、学校教育、社会教育を含む広い意味での教育のゴールを明示することが必要ではないか、との認識に到った。

そこで、米国における「Science for all Americans」の刊行(1989年)に倣って、我が国においても、「Science for all Japanese」を策定する必要があると考え、2005年度に科学技術振興調整費を得て、「科学技術リテラシー構築に向けた調査研究」を推進した。その成果を踏まえて、平成18年度(2006年度)から、我が国の「科学技術の智」すなわち「成人段階を念頭において、全ての人々に身に付けてほしい科学・数学・技術に関係した知識・技能・物の見方」を実際に作成することを目的とした「日本人が身に付けるべき科学技術の基礎的素養に関する調査研究」(平成18・19年度科学技術振興調整費「重要政策課題への機動的対応の推進」)を発足させた。

全体として約150名の科学者、教育者、技術者、マスコミ関係者、また科学技術理解増進に従事する人々などが参加した。このプロジェクトの特徴は、学問の枠を超え、さらに、日本の科学技術の現状と歴史、伝統を踏まえて、科学者と教育学者等が協同で行うことであった。また、一般に公開しながら共に作っていくということを大切にし、ウェブサイトやシンポジウムを活用してできるだけ多くの人々が参画することによって、このプロジェクト自体が科学技術リテラシー向上の運動となることを目指した。

まず全体像に取りかかる前に、現在の膨大な科学技術を七つの分野に分けて、それらに対応する専門部会を組織した。この七つの分野は、学問の体系に対応するのではなく、21世紀を豊かに生きるための智として、関わりの強いところをまとめて一つの分野とした。また、近年急速に大きく広がって社会を変えつつある情報学の分野に対応して「情報学専門部会」を設置し、また、人類が存在する環境としての宇宙と地球に関わる分野について検討するために「宇宙・地球・環境科学専門部会」を設置した。また、人間の行動や社会の現象を科学の視点から考えるために「人間科学・社会科学専門部会」を設置した。物質に関わる分野につ

いて統合的に検討するために「物質科学専門部会」を設置した。数学の本質は、認識とコミュニケーションという人間の基本的精神活動にとって重要な知識と考え方であるという観点から、広く数理的な分野について検討するために「数理学部会」を設置した。生物学から人間に関わる医学や保健までを含め、さらに生命倫理も含めて検討するために「生命科学専門部会」を設置した。技術は、特に日本において近年は科学と強く相互作用しながら進展してきたのであり、またかつては、芸術と一体となって生活の中にあった。社会の在り方と関わる面を考慮した技術の在り方を明示するために「技術専門部会」を設置した。

このように、既存の学問あるいは教科の枠組みを超えた新たな智の領域の枠組みを、七つの専門部会の形で提案したのである。

この各専門部会には、多様な分野の科学者、教育学者、メディア、科学技術理解増進に関わる人々などが参加した。また、部会報告書の原稿が出来上がった段階で、部会間で相互閲読を行い、専門ではない分野の報告書の内容について理解できるように、相互に意見交換を行って、最終原稿をまとめる際の参考とした。

科学技術の智の全体像と其中的個々の知識の間の結びつきを明らかにする作業はまだ途上にあり、むしろその作業を今後とも国民的な協同作業として継続して行くことが、日本の「科学技術の智」を定着化し高めて行くために必要であると考えている。

この七部会報告書が、新たな科学技術理解増進運動の指針として、また、国民的な科学技術の議論と関心を喚起する材料として、多くの人々の手に届くことを願っている。

今後は、この報告書のさらなる改良と、科学技術の智の全体像への統合、さらに、定着化に向けた様々な教材と活動の企画を進めたい。是非、ともに科学技術の智の漲る社会を創成して行きましょう。

まえがき

科学技術が対象とするものは、物質からエネルギーへ、エネルギーから情報へと広がってきた。とくに情報を扱う科学技術は、20世紀の後半に急速な立ち上がりをみせたのである。

ごくごく限られた人たちだけが使うことのできる超高価な装置であったコンピュータが、今ではありとあらゆる機器の中に埋め込まれ、有線・無線で互いに連携して動いている。瞬時にして世界中のニュースや記録の中から欲しい情報を探し出してくることができるし、これから出かける先までの安くて速い交通経路を知ることができる。現金を持たずとも買い物ができ、自宅から預金の振替もできれば株の売買もできる。昨日夢物語であったことが今日は現実になる勢いで我々の生活の中に生かされ便宜をもたらしている。

これらは、装置として実現させるための電気電子工学の技術、そして情報を扱う科学技術¹の恩恵である。本専門部会では、主に後者に焦点をあてて、その本質を理解するための原理や仕組みを中心に解説する。

さて恩恵を与える一方で、情報科学技術は大きな社会問題にも関連している。大量の株が瞬時に誤って売買された結果、株式市場が機能停止に追い込まれたり、自動改札装置が一齐に誤作動を引き起こしたりする事件が起きる。便利なはずの電子メールに多量の広告メールが流され、それらを除去するのに多大な時間をとられてしまう。便利なはずのネットワークからウィルスが知らない間にやってきてコンピュータが機能不全に陥る。知らぬ間に蓄えておいたすべての記録が失われたり、流出してしまったりもする。これらは主に情報科学技術が適正に働かなかつたり、誤って・不正に用いられたために生じた問題である。

情報科学技術は、人間が人間固有の能力だと信じてやまなかった「読み、書き、計算する」知能活動そのものを補強し代行してくれる。そしてなお、発展途上の科学技術分野であり、しかも急速な進展を続けているのである。それは、社会のあり方を変え、人間の行動そのものを変えていく。これまでの常識やモラルや社会的な仕組みのままで正しく対処できないことがらも次々と生じている。

こうした変化に流されることなく、情報科学技術を自らの生活の中に上手に取り込み、自らの知的活動が広がることを楽しみ、新しい社会の仕組みに参画していくには、その根幹にあるものを知っておくことが欠かせない。急速に進展し続ける科学技術分野だから、目の前に提供されてくる機能や機器や商品だけを追っ

¹ 多少語感異なるが記述を簡潔にするために、本報告書では「情報を扱う科学技術」を簡便に「情報科学技術」と略記する。

たのでは今日の常識も明日の非常識になりかねない。情報科学技術の根幹にあるものを押さえておけば、次々と現れてくる新しいことがらにも自ら考え学んで対応していく心掛けを失うことなく、この科学技術の発展の中でも心豊かに過ごしていけるに違いない。

本報告書は、そのような情報科学技術の本質とそれにより得られる様々な技術的な恩恵について解説する。良い面も悪い面も合わせて、情報科学技術は社会に対して大きな影響を持つようになったが、どのような技術的な要因がそうさせたのか、についても述べる。

本部会の名称である「情報学」は、情報に関する学問の総称であり、情報に関する様々な研究が含まれている。しかし、本専門部会は、科学技術リテラシーについての提言をまとめるプロジェクトの中の専門部会である。その位置づけにおいて、情報学専門部会では、情報学の科学技術的な側面について議論し、本報告書もそれに沿った形でまとめた。

具体例でいえば、ブログ²の流行を可能にした科学技術について説明し、その科学技術の特性から来る制約や影響などまでは述べる。しかし、ブログという手段により行われる情報交換の社会的な影響やその分析などの議論までは踏み込まない、という考え方で本報告書を取りまとめた。これは広範な「情報学」の一部の分野にすぎないことをお断りしておく。

ここに解説したような情報を扱うための科学技術の根幹のほとんどは、一般の人々の目にふれることなく、日常で意識することはない。しかし、その正しい理解は21世紀を豊かに生きていくためには重要であり、とくに社会や企業を動かしていく人々には必須のことである。本報告書がその理解の一助になれば幸いである。

² ウェブページに日記や個人の見解を載せ、それを人々に公開すること。

要 約

コンピュータが実際の機械として世に出たのが 1940 年代である。それからわずかの間に、コンピュータとその利用技術は、飛躍的な進捗をとげ、我々の生活や社会に大きな影響力を持つようになった。

この急速な発展には、機械としてのコンピュータや通信機器に関する電気電子工学的な進展とともに、情報を扱う科学技術の進展が不可欠であった。情報を扱う科学技術は、情報を生成し、蓄積し、伝達し、加工するという作業のための科学技術である。その基本原理は、情報を 0 と 1 の列で表わす（デジタル化）と、それを単純な演算の組み合わせで処理する（計算化）の二点である。

本報告書は、この情報科学技術の基本原理の意味と意義を解説し、それに起因する特質を示す。また、様々な情報科学技術の基本的な仕組みを解説し、その仕組みからも特質を述べる。そして、これらの特質から、情報科学技術が、このように飛躍的な進化を遂げたのはなぜか、そして、明暗両面で、我々の生活や社会に大きな影響を与えるようになったのはなぜか、を説明する。その上で、情報科学技術の原理と仕組みを知ることの重要性について論じる。情報科学技術の急速な発展にともなう変化や影響に流されることなく、情報科学技術を自らの生活の中に上手に取り込み、自らの知的活動が広がることを楽しみ、新しい社会の仕組みに参画していくには、その根幹にあるものを知っておくことが欠かせない。急速に進展し続ける科学技術分野だからこそ、目の前に提供されてくる機能や機器や商品だけを追ったのでは今日の常識も明日の非常識になりかねない。原理と仕組みに基づく特質を押さえておけば、次々と現れてくる新しいことがらにも自ら考え学んで対応していくことができるのである。

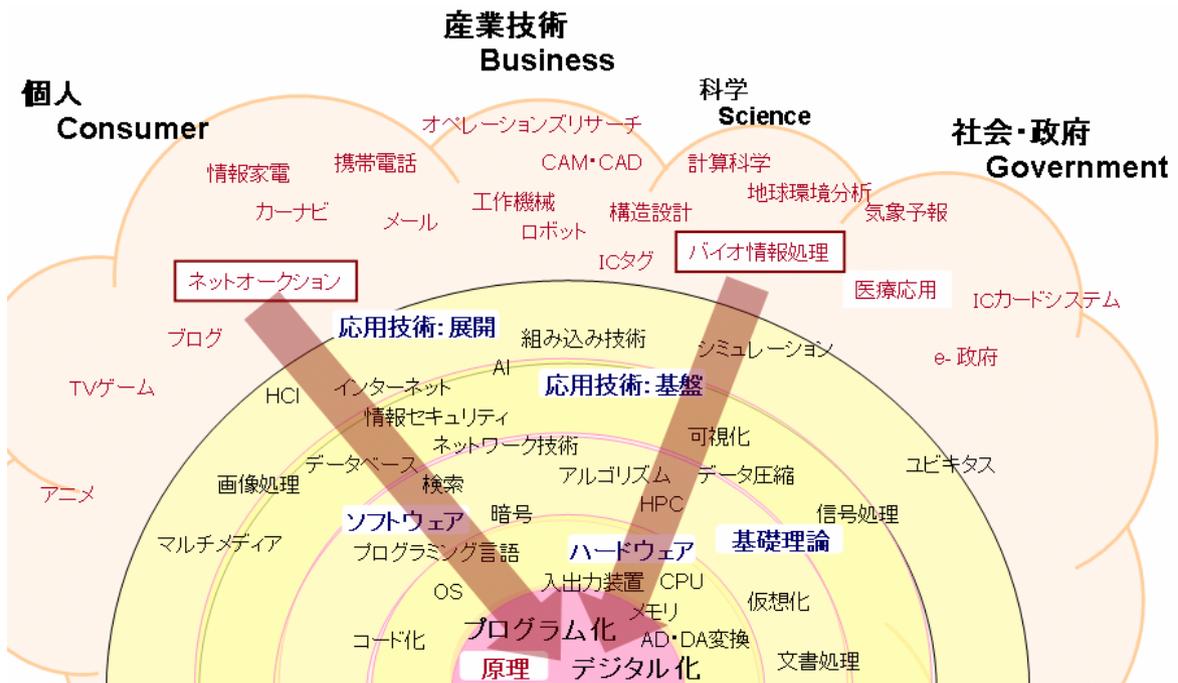
本報告書は以上の点を、科学技術リテラシーの普及を考えている有識者ならびに情報科学技術の教育者を対象に述べたものであり、6 章から構成されている。第 1 章では、報告書全体の概観を述べるために、情報科学技術の本質とそれに起因する特質について述べる。第 2 章では、基本原理（デジタル化と計算化）について、その意味や意義などを詳しく述べる。第 3 章では、情報科学技術の仕組みを述べる。情報科学技術には実に様々なものがあるが、それらに共通する仕組みとその考え方を述べる。さらに、いくつかの重要な技術について、その仕組みを解説する。第 4 章では、さまざまなデータをデジタル化し計算することの利点や問題点について述べる。第 5 章では、情報科学技術の社会的な影響について、情報科学技術のどのような特質が、その原因になっているのか、という点を中心に、事例紹介の形で述べる。そして最終章である第 6 章では、それまでの議論をまとめる形で、なぜ情報科学技術リテラシーを学ぶことが重要なのかを、社会のため、と個人の生活を豊かにするため、の二つの側面から述べる。

情報科学技術ロールケーキ

情報科学技術における基礎理論や重要な要素技術と関連分野の関係を示した図である。ケーキ内部に本部会報告で対象としている情報科学技術の各項目を、周囲のクリーム部（雲のような部分）には、広い意味で情報分野に入る応用領域の項目を配置した。

原理的、基本的なケーキの中心から外側に向かうにつれ、応用領域や他分野との境界領域の研究・技術項目となる。

矢印が書かれているところでは、外側の項目から中心の原理へ、矢印上に深く関係する要素技術が現れるように配置してある。（ただし、この関連付けが正しくなっているのは矢印上のみである。）



略語について

HCI = Human Computer Interface、 AI = Artificial Intelligence、
 CAD = Computer Aided Design、 CAM = Computer Aided Manufacturing、
 HPC = High Performance Computing(Supercomputing)、
 AD ⇔ DA = Analog ⇔ Digital

目 次

「科学技術の智」プロジェクト専門部会報告書の刊行にあたって	i
まえがき	iii
要 約	v
情報科学技術ロールケーキ	vi
目 次	vii
第1章 情報を扱う科学技術の特質	1
1.1 情報処理の三つの本質	1
1.2 情報科学技術とは何か、またその特質とは	2
第2章 情報を扱う科学技術の原理	9
2.1 情報科学技術の基本原則：デジタル化	9
2.1.1 情報とデータ	9
2.1.2 データのデジタル化	10
2.2 情報科学技術の基本原則：計算化、プログラム化	14
2.2.1 計算化への道（その1）：論理演算によるデジタルデータ処理	14
2.2.2 計算化への道（その2）：「繰り返し」の導入	16
2.2.3 計算化への道（その3）：プログラムもデータである！という発見	18
2.3 計算を見極める研究	21
2.3.1 計算可能性の研究：「計算」という概念の明確化	21
2.3.2 真の計算可能性の研究：アルゴリズムとプログラミング	22
2.3.3 計算世界観：計算を中心とした新たな観点	28
第3章 情報を扱う科学技術の仕組み	31
3.1 コンピュータの基本構造：電子機器として	31
3.2 コンピュータの基本構造：ソフトウェアシステムとして	32
3.3 抽象化の観点から見た情報科学技術の仕組み	34
3.4 仮想化の観点から見た情報科学技術の仕組み	36
3.4.1 プログラミング言語の仮想化	36
3.4.2 オペレーティングシステムと仮想化	38
3.5 コンピュータネットワーク	41
3.5.1 情報の享受	41
3.5.2 通信機能の階層化	42

3.5.3	階層化による利点	43
3.5.4	ネットワークの形成	43
3.5.5	インターネット	44
3.6	情報セキュリティ	46
3.6.1	情報セキュリティの重要性	46
3.6.2	情報セキュリティの対象	46
3.6.3	認証	47
3.6.4	データの暗号化	47
3.6.5	共通鍵暗号方式	48
3.6.6	公開鍵暗号方式	48
3.6.7	ハイブリット暗号方式	49
3.6.8	暗号による認証	50
3.6.9	ネットワークに対する情報セキュリティ対策	50
3.6.10	個人・組織・社会における情報セキュリティ対策	52
3.7	基盤となる応用技術	52
3.7.1	データベース	52
3.7.2	データマイニング	53
3.7.3	文書処理	54
3.7.4	検索技術	54
3.7.5	信号処理	55
3.7.6	可視化	55
3.7.7	シミュレーション	55
3.7.8	パターン認識	56
3.7.9	人工知能	56
3.7.10	マルチメディア	56
3.7.11	HCI (Human Computer Interaction)	57
3.7.12	組み込み技術	57
3.7.13	ユビキタスコンピューティング	58
3.8	補足 : 電子回路によるデジタル処理の実現とその発展	58
3.8.1	電子回路の登場 : 論理演算回路の実現	58
3.8.2	電子回路の発展 : 真空管からの脱却	59
3.8.3	メモリ (記憶装置) の進化	60
第4章	デジタル化・計算法の技術的影響	62
4.1	デジタルデータの特徴	62
4.1.1	データの非劣化性	62

4.1.2	データの平等性とメディア非依存性	63
4.1.3	データ処理の高速性	64
4.1.4	デジタルデータの恩恵 :いくつかの例	65
4.2	デジタル化と計算化の問題点	67
4.2.1	デジタルデータは近似にしか過ぎない	67
4.2.2	数値計算には誤差が発生する	68
4.2.3	計算には間違いが入り込む可能性がある	70
第5章	デジタル化・計算化の社会的な影響	73
5.1	見えないデータ・見えないコンピュータ	73
5.1.1	見えないデータ	73
5.1.2	隠されるコンピュータ	74
5.2	ICカードシステム	75
5.2.1	ICカードの仕組み	75
5.2.2	ICカードの役割の変化：偽造防止 → 情報収集	76
5.3	新しい集合知	77
5.3.1	新しい集合知の形成	77
5.3.2	情報科学技術による集合知の光と影	79
5.4	ネットオークション	79
5.4.1	従来のオークション・入札・中古品流通・個人売買	79
5.4.2	ネットオークションの実例	80
5.4.3	ネットオークションによって変わった世界	81
5.5	情報格差 (デジタルデバイド)	82
第6章	なぜ情報学リテラシーなのか	84
6.1	豊かな社会を築くため	84
6.1.1	何を理解すべきか	84
6.1.2	誰が理解すべきか	85
6.1.3	どうやって理解する：情報学リテラシー	87
6.2	豊かに生きるため	87
情報学専門部会名簿		92
「科学技術の智」プロジェクト 研究組織		92
この報告書の利用について		95

第1章 情報を扱う科学技術の特質

情報を扱う科学技術には、いくつかの重要な特質がある。その特質の科学技術的な要因は、情報を生成し、蓄積し、伝達し、加工するという作業（以下では簡単に「情報処理」と呼ぶ）の性質にある。本章では、情報処理の性質の中でも重要なものを「本質」として簡潔にまとめ、その上で情報科学技術の研究・開発というのはどのようなものなのか、さらに情報科学技術が明暗両面で社会的にも大きな影響を持つのはなぜか、なぜ情報科学技術について知っておく必要があるのか、などについても情報処理の本質や情報科学技術の性質と関連して説明する。

以上は本報告書の第2章以降で詳しく述べることの概要でもある。ここでの主張の背景や登場する概念の詳細については、第2章以降を参照されたい。

1.1 情報処理の三つの本質

物事の本質を簡潔に表すのは難しいし、また誤解される恐れもある。しかし、ここではあえて情報処理（情報を生成し、伝達し、加工するという作業）の本質を次の三つにまとめてみる。

本質1：情報を表すデータはすべて0と1で表される。データの加工はすべて単純な計算の組み合わせで表現できる。

本質2：情報の伝達や加工、そして生成の大部分は電子回路により実現できる。

本質3：情報処理の大部分が高速化・自動化できるし、また高速化・自動化される。

これらの本質は独立ではない。本質1→本質2であり、本質1&本質2→本質3である。各本質についてももう少し詳しく述べ、上記の関係について説明する。

本質1：データは0と1で表され、加工は単純な計算の組み合わせで実現できる

情報には本来形がない。人は、それを伝え、記録に残すためにデータという形のあるものにしてきた。そのデータが情報処理の対象となるものであるが、それは0と1の列（これを以下では2進列と呼ぶ）として表すことを本報告書ではデジタル化と呼ぶ。もちろん、近似でしかない場合もあるが、すべてのデータは非常に高い精度でデジタル化できるのである。

一方、そのデータを生成したり、加工したりする処理は、実は、非常に単純な計算の組み合わせで実現可能なのである（もちろん情報処理のうち、情報の生成のように人間の介入が必要な部分もある）。具体的には、デジタル化されたデータを数だとみなすと、それに対する処理は、すべて、

- (1) 自然数に対する+1、-1の計算、そして
- (2) 変数の値が0になるまで指定した作業を繰り返す

という基本計算の組み合わせだけで表わすことができる。このように (1)、(2) の組み合わせで表わすことを本報告書では計算化と呼ぶことにする。

高度な科学技術計算から文書の作成やメールの送受信まで、様々な処理はすべて計算化できる。

本質2： 情報処理は（情報の生成の一部を除き）電子回路により実現できる

本質1の技術的な帰結が本質2である。0と1ならば電気のオンオフ、あるいは電圧の高低で表わすことができる。さらに、そのように電氣的に表わされたデータ（の各ビット）の基本計算であれば、比較的単純な電子回路で実現できる。したがって、本質1の帰結として、すべての情報処理は電子回路により実現できる、という本質が導かれる。ただし、量子計算のような異なるタイプの計算方式や、分子計算のように、電子回路を用いない計算装置も登場しており、従来の電子回路だけではなくなっている。

本質3： 情報処理は高速化・自動化できる。そして高速化・自動化される

情報を伝達したり加工する過程が電子的に実現されることから、情報処理の高速化は当然生まれてくる。また、データの加工（や生成の一部）を実現する時に基本となる「繰り返し」を発展させれば、処理の自動化にもつながる。したがって、本質1と2の帰結として「情報処理は高速化・自動化できる」という本質が導かれる。

さらに高速化・自動化できるのであれば、それを実際に行ないたい（あるいは高速化・自動化へと突き進む）というのが人間の願望であり、社会の要請でもある。この結果、後でも述べるように、大量のデータが世の中にあふれることになり、その量が質をも変える現象を生み出している。

1.2 情報科学技術とは何か、またその特質とは

情報を生成・伝達・加工する作業の本質について整理したが、そうした本質を持つ作業を行なうための情報科学技術の研究とはどういうものなのか、そして研究の成果としての情報科学技術³にはどのような特質があるのかについて説明しよ

³ 「情報科学技術」といった場合、普通は研究の成果（システムやソフトウェア、あるいは理論的な成果）を意味する。ただし「情報科学の研究・開発」という行為と区別したい場合には、「成果としての情報科学技術」ということにする。

う。

デジタル化と計算化の手法

情報科学技術の研究は、ひとことで言ってしまえば、いかにして様々な情報をデジタル化し、また様々な情報処理を計算化するか、という手法の研究である。

「コンピュータに載せる」という言い方があるが、非常に大ざっぱに言えば、そのための研究である。

2進列（0と1の列）という非常に単純なものすべての情報を表わそうというのは単純なことではない。単純な基本計算だけですべての処理を表わすのも同様である。実は「単純でない」には二つの側面がある。やり方が難しい、という面と、表わし方が難しい、という面である。例を二つ挙げて具体的に説明してみよう。

例1：デジタル化・計算化の難しさ（やり方の難しさ）

デジタルカメラで撮影した100万枚の写真データの中から、指定した人がよく撮れている写真を選び出す、という情報処理を考える。人の指定には、その人の写真（これもデジタルカメラで撮ったもの）を与えることにする。

この場合、一番難しいのは「よく撮れている」の計算化だろう。どのような基準にすれば計算化することができ、また多くの場合満足のいく結果が得られるかを考えることである。また、指定された人物の写っている写真自体を探すこと自体も単純ではない。デジタルカメラのデータは2進列である。その2進列をどのように走査すれば対象の人物（これもデジタル画像で与えられている）を探すことができるのか、その方法を考えなければならない。

さらに、同じ100万枚の写真データに対し、この選び出すという情報処理を1000人の人に対して行なう必要があったとする。その場合、100万枚の写真データをそのまま0と1の列で持つておくのは無意味かもしれない。人の探索と「よく撮れている」の基準の計算を効率よく行うためには、元の写真データを別の0と1の列で表わしておいた方がよいかもしれない。効率のよい処理をするためには、デジタルデータとしての表現の仕方自体も目的に応じて考え直してもよいのである。目的のためには「やり方」を選ばない、それだけに新しいアイデアが重要になってくるのである。

例2：デジタル化・計算化の難しさ（表し方の難しさ）

六法全書をコンピュータで利用できるようにすることを考える。いわゆる「コンピュータに載せる」ということである。

もちろん、六法全書にある条文をそのまま文字列として（そしてそれを0と1の列に変換して）デジタル化することもできる。しかし、それでは利用面では紙に書かれたものと大差がない。検索などが容易なように構造化しなくてはならな

い。

ただし、六法全書のように複雑な対象の場合、どうしても人間が構造化をしていかななくてはならないだろう。その場合、かなり複雑な作業になるため、間違い、あるいは、誤解による誤った構造化を防ぐ工夫が必要である。どのようにすれば複雑な構造を、人にわかりやすく表わすか、が重要となってくるのである。また、人での作業の誤りを減らすための補助システム、検査システムの開発も必要だろう。

本質的に複雑なものを複雑なまま（場合によっては、利用しやすいようにさらに複雑にしつつ）デジタル化し計算化するための、表現法や支援方法が重要となってくるのである。

こうした情報科学技術の研究、そして得られる成果としての情報科学技術の特徴の一つに汎用性がある。そもそもコンピュータ自体、非常に汎用である。たった1台のパソコンで、ワープロにも、通信機器にも、ちょっとした科学計算にも利用できるのである。

この汎用性は、すべてのものが究極的には2進列という同じもので表わせること、そしてすべての情報処理が同じ基本計算の組み合わせでできることにも関係している。あるデジタルデータに対して有効な方法が、別のデジタルデータに対しても有効である場合が多々あるからである。

たとえば、2進列の中に与えられた2進列のパターンを高速に探し出す計算方法（パターン検索アルゴリズム）を発見したとする。それは先の例の写真の検索にも使えるかもしれないし、六法全書の中の条文の検索にも使えるかもしれないのである。

この考え方は、情報科学技術の特徴のもう一つの特徴である階層化につながる。

原理的には単純な基本計算の組み合わせですべてが実現できるのだが、すべてを基本計算から作り上げると非常に複雑になってしまう。そこで上のパターン検索アルゴリズムのように、多くのシステムで共通に使えるものを部品として作っておき、それらをあたかも基本計算のようにして新たなシステムを作るのである。さらに高度な部品を作ることも多い。

このような部品群の集まりが階層をなす。基本計算から直接作り上げた部品群、さらにそれらを元に作り上げた部品群、という階層である。

ここで大切なのは、上位の階層では下位の階層で作られた部品の仕組みが重要ではなく、どんな処理結果を出すか、という仕様だけが重要であるという点である。つまり、同じ処理結果を出す部品ならば、いつでも別の部品に入れ替えることができるという点である。この考え方は（処理の）抽象化と呼ばれている。

部品の仕様を明確にし、抽象化を確実にしておけば、同じ仕様を満たす部品と

の組み合わせが自由にできる。それが様々な情報科学技術の開発に大きな自由度を与えているのである。たとえば、従来と同じ仕様を満たす部品で、さらに従来にない機能を持たせた部品を提供することで、これまでのシステムも同時に生かしながら、新たな利用法も提供できるという拡張ができる。これが情報科学技術の変化の速さの一因にもなっている。ただし、仕様が合わないために、優れた仕組みでも（すぐには）利用できないという状況も比較的頻繁に起きている。

大量のデータ処理

情報処理の高速化と自動化は大量の（デジタル）データを生み出すことにつながる。情報科学技術の重要な課題は、こうした大量のデータをうまくさばき、我々にとって有益なデータを作り出すことである。そのためには、電子化に伴って得られる単純な高速化や自動化だけでなく、より高度な高速化やデータの自動処理の仕組みが必須となる。それを研究するのが情報科学技術なのである。

情報科学技術の「自動」という考え方の元には再帰という考え方がある。これは情報科学技術の自己増殖性という特質を生み出す。簡単にいえば、放っておいても自ら拡大していく性質である。したがって、情報科学技術である高度な高速化やデータの自動処理の仕組みは、さらに多くのデータを作り出すことにもなる。つまり、大量のデータをさばく技術は、同時に、大量のデータを作り出すという結果にもつながるのである。

例 1：高度な高速化や自動処理の仕組みの不可欠

Deep Blue だけではチェスは勝てない

1997年にディープ・ブルーと呼ばれるスパコンがチェスの対決において、人間の世界チャンピオンに勝利した。

スパコンは最先端の電気電子工学と科学技術の成果である。その計算力が勝利の鍵だったことは確かだが、機械だけでは一流のチェス選手にすらなれない。

ディープ・ブルーは1秒間に2億手の先読みを行うことができたが、単純にすべての手を読んでいたのでは、そのスピードでも深読みできない。勝つためには何が必要かの目標を巧妙に定め、その目標を高速に達成するための高度な探索アルゴリズムが必要だったのである。

大量データだけでは知識にならない

現在、インターネット上に極めて大量のデータが存在する。Googleのような代表的な検索エンジンは、その膨大なデータを利用して知識を提供することに成功しており、「Google先生」という言い方さえ生まれてきている。しかし、大量にデータを集めただけでは、人々に知識を与えることはできない。多種多様、種々雑多なデータの中から、利用者の知りたいことに近い情報を自動的に選別し、場合によっては加工することで、初めて人にとって役に立つデータを提供できるのである。

電子的な高速化の上昇傾向は徐々にではあるが頭打ちになりつつある。しかしそれに変わって情報科学技術は通信技術の向上と共に大量のデータの流通をさらに推進化することになるだろう。

情報科学技術の影響：なぜ情報科学技術リテラシー

情報科学技術が社会にここまで影響を及ぼすようになったことの一次的要因は、情報処理が電子化できたこと、そして、様々なデータを高速に通信できるようになったことである。両者とも電気電子工学の成果といえるが、情報科学技術自身も重要な役割を担ってきた。たとえば、2進列化されたデータを分解して高速に送るといった情報科学技術的な発想がデータの高速通信を実現させた。

そして二次的要因は、情報科学技術の研究の成果として複雑で大量のデータを自動的に処理する様々な情報科学技術が提供されたことである。人々がそれらを用いて比較的安易に、大量のデータを作り出し交換するようになったためである。情報科学技術の自己増殖性という特質は、良きにつけ悪きにつけ情報科学技術が大きな影響を与えるもとになっている。たとえば、多くの人に同報通信ができる特徴を持つ電子メールは、緊急時の連絡のみならず日常でも大いに役立っている。しかし一方で、自己増殖性はコンピュータウイルスが爆発的に広まる原因にもなり得る。

こうした情報科学技術について、その原理や仕組み、そしてそれらの技術的な意味を知っておくことの必要性について、これまで述べてきた情報処理の本質や情報科学技術の特質から確認してみよう。

まず、情報科学技術のほとんどが目に見えないところで働いているという点がある。具体的には電子回路などで実現されているが、情報科学技術はその中で動いている目に見えない物である。さらに情報科学技術の**変化の速さ**である。これは情報科学技術がまだまだ成長段階にあるためでもあるが、情報科学技術の特質であるとも考えられる。

残念ながら、目に見えない物に対して、その仕組みや特質を見抜くことのできる人は少ない。しかも目に見えるところだけを見ていただけでは、その変化の速さに惑わされる結果になってしまう。変化の速さや量の効果の影響を見過ごし、情報科学技術に対する誤った判断をもたらす可能性がある。

行政、企業、教育において責任ある地位にある人々が、こうした誤った判断をしてしまうと致命的な問題を生み出すことがある。こうした人たちが、実際に情報科学技術を使って情報処理システムを作るわけではない。しかし、どのような情報処理システムが必要か、という判断を下したり、でき上がった情報処理システムを運用していく際の規則や制度設計などを行うことは多いだろう。その際に、

情報科学技術に対する誤った判断をした場合、それが社会に与える影響は非常に大きい。

個人レベルでも、情報科学技術に対する誤解が、その個人だけでなく、多くの人々へ被害を与えることもある。情報科学技術の高速性や自己増殖性は、個々の影響力をこれまでよりはるかに大きくしてしまう場合があるからである。

こうした「知らねば困る」という点がある一方で、情報科学技術には「知っている」と豊かに暮らせる」という面も多くある。情報科学技術は単純な操作の組み合わせであり、多くの場合、それを修正・変更することも可能である。つまり、その仕組みさえわかれば、誰もが自分流に改変することが可能なのである。情報科学技術の速度の速い変化に惑わされることなく、自分流に情報科学技術を使っていくすべを得るためには、そう多くのことを学ぶ必要はない。

本部会報告で述べるのは、そうした情報科学技術に関する基本なのである。

仮想の疑似通貨を使った出費をめぐる犯罪が摘発されている。現実の通貨で出資すると同額の仮想通貨が渡され、それを使って現実の物品を購入する。この仮想通貨を使い切ると、同額の仮想通貨が何回でも支給される。単純に考えると、初回の出資のみで無限に買い物ができることになる。こんなうまい話はないが、どこに落とし穴があるのだろうか。この状況を理解するためには、(数学ではなく)情報学的素養が必要である。上記の犯罪に対する疑問は、このシステムを数学的に、すなわち静的に把握することから発生する。これに対して計算を基本とする情報学の考え方によれば、システム全体は動的に計算されるのであり、静的なビューでは把握できない、この犯罪の実施側の最適プランは、実施側の身入りが最大になった時点で事業を中止することである。もちろん、中止した場合の社会的費用も考えに入れる必要がある。

情報学リテラシーは、数学リテラシーが与える静的なビューを格段と豊富にするのに大いに寄与する。この世の中の事象は、すべての条件設定をあらかじめ行なってから解いた解答として動いているのではなく、解となるシステムを取りまく環境が動的に変化し続けている中で発生し進行しているのである。しかもこれらの環境が情報システムに依存する度合はますます増大しており、それについてわれわれの生活自体もある種の仮想環境を前提とするようになってきている。「仕事がうまくゆかないよ、エイッ、リセットできないなあ」というCMがある。また「本物の喧嘩をしたことがないので肉体的な痛みは無知であることから、相手に深いダメージを与えてしまう」という議論がある。確かにその種の事件が多発している。これらは、我々の日常生活や感覚に、いかに仮想環境が浸透しているかを示している。

我々はもともとは自然環境の中で育ちながら、各種の知覚を発達させるとともに、その中で「うまく」生活するための知識や思考方法に親しんできた。多くの自然科学あるいは社会科学的な「リテラシー」はこのようにして醸成されてきたと言える。ひるがえって仮想環境についてはどうであろうか、システムの開発者達は、一般人を対象とする各種の仮想環境について、人ができる限り自然に接することができるようなインタフェースを作成するように努力する。しかし所詮は「作りもの」であり、長大な時間を経ている自然環境にかなう筈もない。さらに、情報システムの発達速度が自然及び社会システムのそれに比べて異様なほどに速いという状況もある。そのような状況下では、仮想環境に全幅の信頼を置き自分では何も考えないというのではなく、その成り立ちについての基本的かつ普遍的な知識をもち、様々な状況に対処できる力を身につけておくことが必要である。これが情報学リテラシーであり、いわゆる「無知であることから (知ることによって)自由になれる」というリベラルアーツそのものなのである。

第2章 情報を扱う科学技術の原理

情報を扱う科学技術は、基本的な二つの重要な原理の上に成立しているといってもよい。「情報」を0、1の列としてデジタル化して処理すること、そして、その処理を単純な操作の組み合わせの「計算」としてとらえること、すなわち計算化（プログラム化）である。

デジタル化と計算化は単純な原理である。しかしながら、実態のつかみにくい「情報」をデジタル化し、自然科学から日常生活まで多岐に渡る様々な事柄を「計算」としてとらえるためには、重要な発見や理論、様々な要素技術が必要であった。情報科学技術が今日、ここまで発展したのは、そうした発見や開発の積み重ねであったといってもよいだろう。

この章では、デジタル化と計算化という原理とその理論的な重要性を紹介するとともに、この二つの原理を可能にした重要な理論や技術について述べる。また、計算という新しい見方の重要性についても説明していく。

2.1 情報科学技術の基本原則：デジタル化

2.1.1 情報とデータ

情報とは、人が人に伝えたい、あるいは伝えると何らかの意味を持つ「こと」である。ある人が距離を隔てた他の人に伝えたい場合もあれば、何らかの記録やメモのように、時間を越えて自分自身に伝えたい場合もある。

情報には本来は形がない。けれども伝えるという目的のもとに、人間は情報に形を与えようと努力してきた。その努力の結果、多くの情報は形を持つようになった。文書などはその代表例であるが、絵画や写真として紙の上に表されたもの、ビデオや録音によってCD上に記録されたものなど、さまざまな形で情報は表されている。それらを総称して本報告書では（広い意味での）データと呼ぶことにする。

このようにデータという形になった情報を扱う科学技術が本報告書の主な対象である。ただし、広い意味のデータを考えたとしても安易に「情報＝データ」と仮定してはいけない。という点をここで注意しておこう。

まず第一に「データだけでは情報にならない」という点である。厳密に言えば正しい解釈のもとでデータを見なければ情報は得られないのである。たとえば、CDに記録された音楽も、正しい復元の仕方をしなければ聞くことができない。日本語で書かれた文を日本語がわからない人が見ても情報はほとんど得られない。また、たとえ日本語がわかったとしても、正しい解釈のための十分な知識がなければ、書き手がどんな情報を伝えたかったのかがわからないだろう。このように、

解釈にも機械的なものから、完全な規則にしにくいものまで、さまざまなレベルがある。こうした解釈のもとで、データが何らかの情報を表わしていることになるのである。

第二は「データになったものだけが対象ではない」という点がある。ある時点で、データになりえないと思われている情報も、何とか形（データ）にしようという研究も行われている。たとえば、においや味などの情報や、運動感覚などもデータにしようという試みが行われており、これらも情報を扱う科学技術の研究の一部である。

以上の注意をふまえたうえで、以下では、主に形（データ）になった情報を扱う科学技術について述べていく。

2.1.2 データのデジタル化

情報を形にし、記録や複製、そして伝達するための手段として、人間は古代から象形文字のような記号を用いてきた。それは近代では文字となり、文字を連ねて情報を文章として表現するようになった。

どの言語でも、文字が決まっていて、それを並べることで情報を表現する。我々は、その有限種類の文字を並べることで情報を表現し、その表現されたものを記録し、伝送し、編集したりすることで、情報の記録・伝達・処理を行っている。この方式をとことんまで突き詰めたのが、今日の情報科学技術の根幹となるデジタル表現である。

デジタル表現は様々な意味で用いられているが、本報告書での意味を明確に規定しておく、まず、0と1の列を2進列と呼ぶことにする。2進数はビット列ともいう。2進列は、文字の種類を0と1の二種類に限定した文字列である。我々はデータを、この2進列で表すことを、データの2進表現、あるいはデータのデジタル化⁴と呼ぶ。

では、どのようなデータが2進表現できるのだろうか。この問いに対し次のように答えるのが、情報科学技術の基本原理の1つであるデジタル化の原理である。

デジタル化の原理

すべてのデータは2進列で表わすことができる。

⁴ 0～9の数の列で表わすこと、あるいは何かの記号列で表わすことを「デジタル化」と呼ぶ場合も多い。しかし、2進表現の重要性を示すために、本報告書では、あえて2進列で表わすことをデジタル化と呼ぶことにした。

原理と称したが、これは自然界の法則のように自然に成り立っているものではない。この原理に向けて、さまざまな情報科学技術が駆使され、その成果として実現され、実証されてきた原理なのである。そのデジタル化の事例をいくつか紹介しよう。

数字や文字のように、もともと記号になっているデータは比較的簡単に 2 進列にできる。そのように、もともとは記号になっていないデータ、たとえば音や絵のようなデータをアナログデータという。

こうしたアナログデータもデジタル化できる、つまり 2 進列にできる、という発見は、デジタル化の原理を実現するための重要な一歩であった。(本文中では、省略するが、アナログデータを 2 進列で表わす例として以下の例で、音と画像のデジタル表現法を紹介するので、興味のある方は参照されたい。)

用語解説1: ビット、バイト

2 進列の構成要素である一つ一つの 0 や 1 をビットという。そのため 2 進列をビット列ともいう。

ビットはまた、2 進列の長さ (桁数) の単位でもある。たとえば、4 ビットの列とは、0101 のように 4 個の 0 または 1 からなる 2 進列のことである。

また 8 ビットの長さを 1 バイトといい、これも長さの単位によく用いられている。基本的な文字には 1 バイトの 2 進列が割り当てられている。

例2: 数字や文字の2進表現

数 (自然数) は 2 進数という形で表わすことができるので 2 進表現可能である。たとえば、6 は 110 で、25 は 11001 と表わされる。

$$\begin{array}{r} 1 \ 1 \ 0 \Rightarrow 6 \\ \uparrow \ \uparrow \ \uparrow \quad \uparrow \\ 4 \ 2 \ 1 \quad 4 + 2 \end{array}$$

10 進数では、1 の位、10 の位、100 の位、... で表わしていたのに対し、2 進数では、1 の位、2 の位、4 の位、... で表わすのである。

$$\begin{array}{r} 1 \ 1 \ 0 \ 0 \ 1 \Rightarrow 25 \\ \uparrow \ \uparrow \ \uparrow \ \uparrow \ \uparrow \quad \uparrow \\ 16 \ 8 \ 4 \ 2 \ 1 \quad 16 + 8 + 1 \end{array}$$

負の数や小数 (ただし有限桁) を表わすには、基本的には二つの情報が必要である。

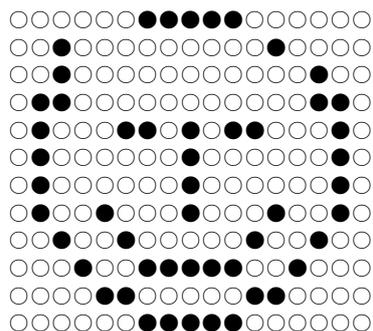
たとえば、+2 と -2 を表わすには、2 という絶対値の他に、正負の符号を表わす必要がある。小数は、たとえば $0.25 = 25 \times 10^{-2}$ と書けることを利用すれば、二つの数 (25 と -2) で表現できる。ただし、現在のコンピュータでは、ある程度の大きさの数 (たとえば 2^{31} = 約 21 億) までを簡便に扱える表現を用いている。具体的には、負の数は補数表記で、小数は浮動小数点表記という方法で表わしている。

文字の場合には、各文字に適当な 2 進列を割り当てるコード表を決めておけば、2 進列で表わすことができる。たとえば、基本的な記号と英字アルファベットには 7 桁の 2 進列を割り当てる ASCII コード (アスキーコード) が一般的に用いられる。

例3:画像を2進列で表わす

新聞の写真をよく見ると細かな点が印刷されているのがわかる。画像を白と黒の点で表わすことができるのである。

たとえば、下図（左）は白点○と黒点●を12行17列並べて、絵を描いたものである。これではとても粗くて精密な絵は描けないが、右（256行256列）ならば、かなり絵らしく描ける（注：実際には白点○ではなく見えない点が印字されるので、結果とし図右のように黒点だけの印字となる）。このように白（空白）と黒の印字の指示であれば1ビット（0は黒、



12×17



図 ●で絵を描く

1は空白)で指定できる。つまり、左図は $12 \times 17 = 204$ ビットの2進列で右図は $256 \times 256 =$ 約6万5千ビットの2進列で表わすことができるのである。

カラーの画像はどうだろうか？ ノートパソコンの液晶ディスプレイを例にとって説明しよう。こうした液晶ディスプレイ上には非常に小さい画素（ピクセル）が規則的に並んでいる。一つの画素は、赤、緑、青の3個の光源から成る。直感的には赤、緑、青三つの電球（各々、R、G、Bと呼ばれる）が○のように並んだものが1画素で、それが、たとえば、480行640列並んだものが液晶である。（ちなみに、写真などのように紙へ印刷する場合には4色が基本色として用いられる。）

デジタルカメラの精度などで「25万画素（ピクセル）」などというキーワードが出てくるが、これはその画素数のことである。たとえば（白黒画像だが）、左の図は $12 \times 17 = 204$ 画素、右図は $256 \times 256 =$ 約6万5千画素である。

液晶ディスプレイの一つの画素に注目してみよう。それを構成する赤、緑、青の三つのランプの輝度（光の強さ）はディスプレイでは、それぞれ0（光らない）から255（最も明るい）まで、その明るさを256段階に分けてコントロールできるようになっている。すべてが0のときは、その画素は黒く見え、すべてが255のときは白く光って見える。その他の組み合わせで様々な色を表わすことができる。たとえば、赤、緑、青の明るさが255、0、255では紫、255、255、0のときは黄色に見えるのである。

255という数は、2進数8ビット（8桁）で表わすことのできる最大数である。つまり、一つの画素の色は8ビットの数3個（24ビット）で表わされている。一つの（液晶などの）画面は、この24ビット×画素数のビット数で表わすことができるのである。

一般にアナログデータのデジタル化は近似であり、完璧な精度は望めない。しかし、たとえば画像では、画素数や階調（色の濃さの度合いの段階数）を多くすれば、それだけ滑らかに画像を表すことができる。音の場合も刻み幅をとる周期を増せば精度が上がる。このように、理論的にはいくらでも精度を上げられる、という性質は、デジタル化の方法を考える際に必須の要件である。

デジタル化では原理的に精度をいくらでも上げられることができるような表記法が用いられる。けれども現実には有限の精度内で表わさなければならない。そのような場合に精度は妥協の結果決められる、という点も重要である。

つまり、いくらでも精度を上げることは可能だが、その時点での技術レベルや生産コストと人間の要求とが妥協点に決まるのがほとんどである。妥当な値段で提供できて人間が十分満足できるよう設定されるのが普通なのである。

用語解説2：精度の単位 dpi、Hz

画素数は画像の解像度を表わすのに使われているが、本来は「数」ではなくて密度が重要である。たとえば、1000画素といっても、縦横10cmなのか10mなのかで精度が大きく異なってくる。画像の場合に密度を表わす単位は、dpi (dot per inch) つまり1インチ当たりの点（画素）の数である。

音の場合には、その標本値を取る頻度が精度の指標の一つになる。この頻度は周波数の単位であるHz（ヘルツ）で表わす。たとえば44.1kHzは1秒当たり44100回。1/44100秒刻みで標本値を取るという意味である。

例4：音を2進列で表わす

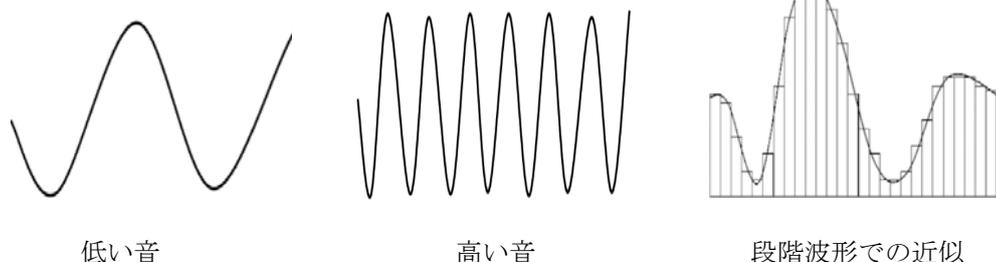


図 音（波）をデジタル化する

音は電気信号として、時々刻々変化する電気の強さ（電圧）で表わされる。直感的には上図のような波である。横軸が時刻、縦軸がその時刻での信号の強さである。音の高低は波の山の多さ（単位時間あたりの振動数）で決まる。一方、波の高さは音の大きさである。

この波はアナログデータである。それをデジタルで表わす基本は階段状の棒グラフでの近似である。ある適当な刻み幅で上図右のような棒グラフ化し、その高さ（標本値という）の列で波を表わすのである。刻み幅が小さいほど、標本値を精密に表わすほど、近似の精度が上がる。とくに、振動の激しい高い音を表現するにはそれに応じた刻み幅が必要で、CDなどでは普通、44.1kHz（1秒間に44100回）の刻み幅が用いられている。

アナログデータをデジタルデータとして表わす表現方法を決めたとしても、実際にはその表現法に基づき、アナログからデジタルへ、またデジタルからアナログへ、変換しなければならない。前者をAD変換、後者をDA変換という。このAD・DA変換もデジタル化の基礎技術である。

2.2 情報科学技術の基本原則：計算化、プログラム化

2.2.1 計算化への道（その1）：論理演算によるデジタルデータ処理

すべてのデータを究極的には0と1の数字からなる2進列で表わすことができる。というデジタル化の原理は、実に様々な効用を我々にもたらしてくれる。その最も重要なものがデータ処理（計算）の単純化である。

すべてのデータを2進列で表わすことで、すべてのデータ処理を非常に単純な基本操作に分解できる。2進列の構成要素は0と1である。そこで処理の基本は、この0と1の計算規則（論理演算：解説1参照）に集約できるのである。

基本的な論理演算は、二つ（または一つ）の1ビットのデータから1ビットの答えを求める計算である。このような単純な計算であれば、電子素子で安定して実現することができる。これが今日の情報科学技術の第一歩となる発見である。1ビットのデータであれば、電気信号のオンオフで表わすことができる。そのような1ビットの信号を二つ入力し（各 x_1 、 x_2 とする）、たとえば、論理積 $x_1 \wedge x_2$ を1ビットの信号（これを y とする）として出力する素子を実現可能なのである。このような素子を（論理）演算ゲートあるいは（論理）演算素子などという。

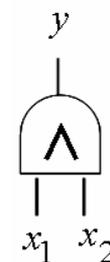


図1 論理積ゲート

こうしたゲートを図1のように表わすことにする。本来は、信号の入ってくるタイミングなどを調整する必要があるが、以下では話を簡単にするため、そのような点は無視し、データは瞬時に処理されるものとする。

解説1：論理演算

0と1の計算を論理演算という。「論理演算」とは、本来は真と偽の二値に対する計算であるが、1を「真」、0を「偽」と考えると、0と1の計算は論理値間の計算とみなせるからである。以下に示したのは、それぞれ論理積（ \wedge ）、論理和（ \vee ）、否定（ \neg ）の計算規則である。 \wedge はAND演算とも呼ばれる。 $x \wedge y$ が1（真）となるのは x と y の両方が共に真の場合のみだからである。同様な理由で \vee はOR演算、 \neg は否定演算と呼ばれている。

$$0 \wedge 0 = 0 \wedge 1 = 1 \wedge 0 = 0, \text{ そして } 1 \wedge 1 = 1$$

$$0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1, \text{ そして } 0 \vee 0 = 0$$

$$\neg 0 = 1 \text{ そして } \neg 1 = 0$$

この三つは論理演算の基本である。

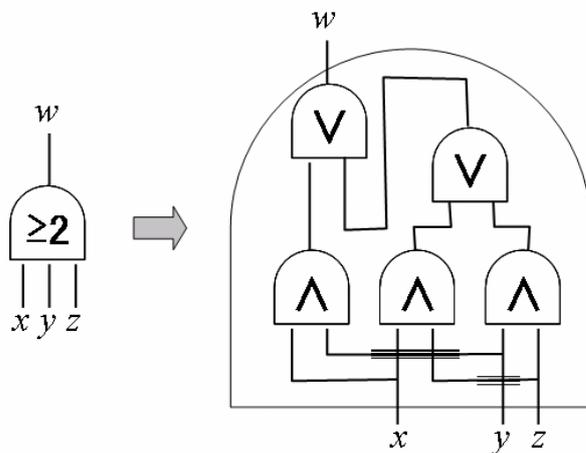
図 1 は論理積を計算する素子であるが、その他に、論理和、否定を計算する論理演算ゲートさえあれば、すべての計算を実現することが可能である。

たとえば、次のように定義される演算 \oplus (排他論理和) を考えてみよう。

$$0 \oplus 0 = 1 \oplus 1 = 0 \text{ そして } 1 \oplus 0 = 1 \oplus 0 = 1$$

この論理演算も、 $x \oplus y = (\neg x \wedge y) \vee (x \wedge \neg y)$ という一般式が成り立つので、図 2 のような組み合わせで計算できる。

さらに複雑な計算も、こうした演算ゲートを組み合わせれば実現できる。たとえば図 3 のように、基本演算ゲートをうまく配線すれば、より複雑な演算を行うことができる。



入力 x, y, z のうち、1 が 2 個以上あるか否かを判定する計算を実現する方法。配線が交差しているときは太線が上で、その下を細線が通っているものとする。

図 3 より複雑な計算

実現することができる。この種の回路を組合せ論理回路という。(例 5 に足し算を行なう回路例を示したので興味のある方は参照されたい。)

組み合わせ論理回路により、2 進列の各ビットを入力とし、それらに対し何らかの処理をしたものを計算することができる。それらに対し何らかの処理をしたものを計算することができる。これは足し算や掛け算のような「計算」に限らない。画像や音声のデータ処理、あるいは、文字や言葉の変換も同様に可能なのである。

情報科学技術の大きな特徴は「計算」のこの汎用性にある。データは 2 進列という同じ顔を持ち、その処理は同一の論理演算の組み合わせで行なうことができる。そのため、同じ演算回路を様々な用途に用いることができ、共通の部品を利

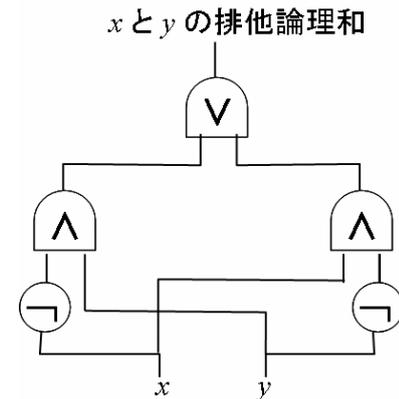


図 2 排他論理和の計算

さらに、この図 3 の回路全体を入力が 3 個の一つのゲートのように見することもできる。(ここでは、 ≥ 2 という名前のゲートとみなしている。) このような回路を部品として使って、さらに複雑な計算を行うこともできるのである。

こうして順次組み上げていけば、どのような複雑な計算も原理的には

用することができ、それらを量産することができるのである。つまり、情報科学技術を支える部品は量産することができ、そのため焦点を絞った技術革新の競争が激しく行われてきたのである。その結果、記録や処理できるビット数がとてつもなく増えた。実際、回路は非常に小さくなり高速になった。半導体を使った電子回路技術の進展によって、量も速度も3年に2倍（おおよそ、10年で10倍）の勢いで増えてきたのである。

2.2.2 計算化への道（その2）：「繰り返し」の導入

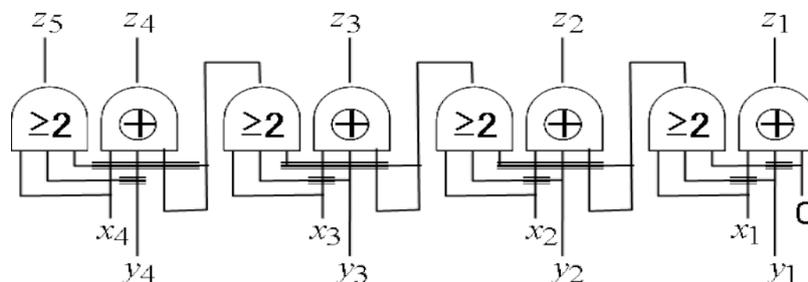
論理演算の組み合わせで、すべてのデジタルの処理が可能、というのは計算化への大きな一歩であるが、計算化にはまだ重要な要素が二つある。

演算ゲートの組み合わせである組み合わせ論理回路だけでは、決まった大きさのデータしか扱うことができない。たとえば、下の例に示した組み合わせ回路では4ビット同士の自然数の足し算しかすることができない。もちろん、同様の方法で、10ビット、50ビット、あるいは100ビットの自然数の足し算の回路を設計することはできるが、ビット数の増加とともに、毎回、新たな論理回路が必要になってしまう。こうした問題を解決する方法が「繰り返し」である。

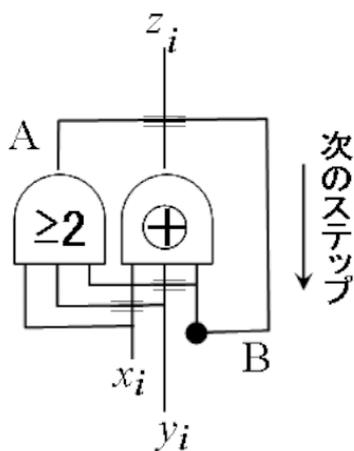
計算では、似たようなことを繰り返し行うことが多い。足し算を計算する回路を見ても同じパターンが繰り返されているのがわかるだろう。これは各桁 x_i と y_i 、そして下からの繰り上がり値の和を求め、その1桁目を対応する桁の答えの値 z_i とし、2桁目を新たな繰り上がりの値とする。これを繰り返しているのである。

例5：足し算を行なう回路

2進列のデータを処理する回路の例として、4ビットの2進数の和を求める回路を図6に示す。計算したい4ビットの2進数の各桁が、各々 x_4, x_3, x_2, x_1 と y_4, y_3, y_2, y_1 に与えられたとき、その和が z_5, z_4, z_3, z_2, z_1 に計算される。たとえば $x_4x_3x_2x_1=1101$ と $y_4y_3y_2y_1=0110$ を入力すると、 $z_5z_4z_3z_2z_1$ として 10011 が出力される。詳細は省略するが興味のある方は試してみられるとよいだろう。



図中で ≥ 2 という印のついたゲートは図3で定義したものの、 \oplus ゲートは、3変数の排他論理和を行なうゲート。入力中の1の個数が奇数個ならば1、偶数個ならば0を出力する。2変数の排他論理和ゲートを2個組み合わせれば実現できる。



足し算を行なう順序回路の概念図。入力 x_i と y_i に足す数、足される数の i ビット目が、 $i=1$ から順次来ると、 z_i に答えの i ビット目が求まる。それと同時に桁上がり A で計算され、それが次の入力があるタイミングで B に送られて足しこまれる。

図 4 順序回路による加算

したがって、この計算を繰り返すような仕組みができれば、1つの回路でどんな桁数の足し算も、原理的には可能になる。その仕組みを取り入れた回路が図 4 のような順序回路である。

図 4 は、かなり概念的な図であるが、それでも順序回路の仕組みと特徴が表わされている。第一に、本当の入力である n ビットの 2 数、 $x = x_n \dots x_1$ と $y = y_n \dots y_1$ に対して、適当な順序で（この場合には x_1, y_1 からの順で）入力ビットが順次、回路に渡される、という仕組みである。第二に、ある時点で計算されたものを、同じ回路の入力ビットとして与えることができる、という点である。ただし、この場合、現時点での入力ビットではなく、次の時点の入力と同時に与えられる。

図 4 は、こうした仕組みを仮定した概念図であるが、実際にはその実現のために、いろいろな技術が必要である。（以下に興味のある方のために、関連技術の簡単な解説を付け加えた。）

解説2：順序回路のための技術

図 4 のような計算を実際にも実現しようとしたときの一番の問題は、0 や 1 などの入力信号がどのタイミングで来るか？ という点である。このようにタイミングをとることを同期をとるといふ。

現在使われているほとんどの回路では、クロックという仕組みで実際に同期をとるようになっていっている。簡単に言えば、「いっせいの、せつ！」と掛け声をかけてくれる誰かがいて、その掛け声に合わせて信号が次の素子に送られる、という仕組みである。当然、この掛け声かけのタイミングが回路の計算の最小時間になる。そのタイミングの頻度を表わすのにも Hz が使われている。たとえば、1.2GHz というとき、1 秒間に 1.2×10^9 回（12 億回）、掛け声をかけるという意味である。したがって、1.2GHz の回路では、一番簡単な（基本的な）演算ならば、1 秒間に 12 億回できるかもしれないが、それより早い計算はできない。

2.2.3 計算化への道（その3）：プログラムもデータである！という発見

ここまで計算化のために「計算」の基本要素について説明してきた。つまり、基本的な計算を行なう論理演算ゲートと順序回路の繰り返しの仕組みである。この二つの要素を、もう少し抽象的に考えてみよう。

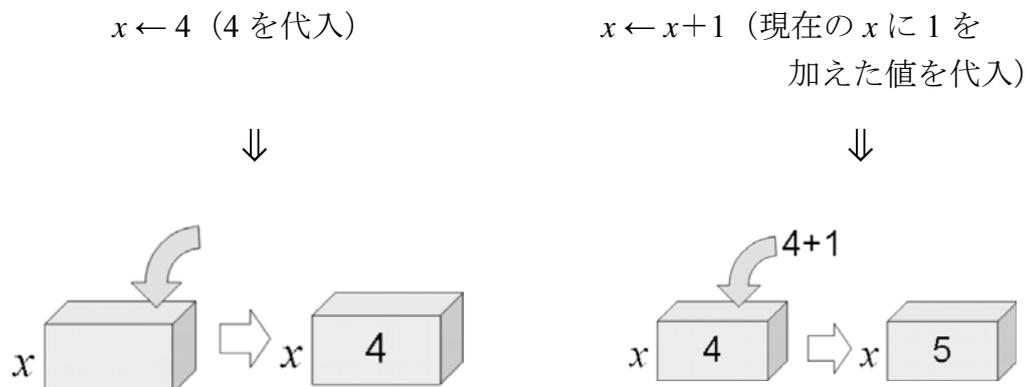
抽象的に「繰り返し」を議論するための鍵となるのが「状態」である。順序回路が組み合わせ論理回路と大きく違う点は、時間経過の概念が入ってきている点である。時間経過は「繰り返し」を議論するために必須である。計算を時間を追って行っていく中で変化するのが「状態」であり、次の時点に計算途中の情報を伝えるのも「状態」である。

具体的に説明しよう。プログラムでは変数という概念が登場する。これは数学における変数とは異なる。データをたくわえておく「箱」と思った方がよいだろう。たとえば、数が一つしまわれている、と考えてもよい。このような箱（変数）をたくさん用意しておき、その値を順次書き換える（上書きする）ことによって、計算途中の値をしまっておくのである。

この変数の値が「状態」である。変数を通して、ある時点の計算で得られた値を「繰り返し」の次の時点に引き継ぐのである。たとえば、足し算の計算を図4のような「繰り返し」で行なうときに、 i ビット目同士の和によって生じる繰り上がりを保持するのが変数である。それが次の繰り返して $i+1$ ビット目の計算に用いられるのである。

解説3：プログラムにおける変数

プログラムでは「変数」という概念が登場する。これは数学で用いられる変数とは異なる。数学では一度決まると変数の値は不変だが、計算の世界では代入という操作で値を変えることができる。変数は計算途中の値をしまっておく箱と考えた方がよいだろう。



変数に格納させる値として、ここでは自然数を用いることにする。その場合には論理値の基本演算に対応するものは自然数に対する基本演算である。ここでは+1、-1の計算だけに限ることにしよう。つまり、変数に格納させている値を1増やす/減らす計算である。また順序回路の「繰り返し」に対応するのは、基本演算の組み合わせを必要なだけ繰り返すことである。ここで「必要なだけ」を指定する方法が必要になってくる。これは「ある指定した変数の値が0になるまで」(変数=0になるまでの繰り返し)という指定のやり方のみを考える。

以上の枠組みのもとで計算化の原理を述べよう。

計算化の原理

すべての計算は、

- (1) 変数に対する +1 または -1 の基本演算、
 - (2) 変数 = 0 になるまでの繰り返し、
- の組み合わせで表わすことができる。

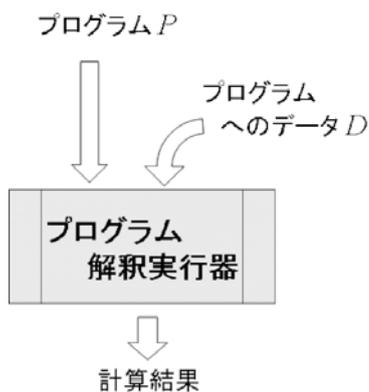
変数に対する±1の計算と単純な繰り返しだけでは、ごく限られたことしかできないように思われる。しかし、組み合わせ論理回路のところでも述べたように、加算ができ、乗算ができる。さらにそれらを組み合わせれば、〇〇ができ、××ができる、… という具合に単純なものから段階を追って考えれば、どのように複雑な計算も原理的には組み立てることができるのである。

なお、ここではデータの入出力は考えていない。実際には、データを(デジタル化して)対応する変数に格納したり、計算結果として各変数に格納されているデータを、文字や画像に変換して出力する部分が必要である。これについてはここでは省略している。

以上のような計算の組み立てを示すのは、回路ならば回路図であるが、それを文章(数式)で明確に記述したのがプログラムである。ただし、通常の記事とは異なり、曖昧さを完全に排除した人工的な言語—プログラミング言語—に従って記述されるのが普通である(第3章参照)。(次ページにプログラムの簡単な例を示した。興味のある方は参照されたい。)

以上、順序回路という形で表現されていた「計算」を、もっと抽象的に考え、プログラムという形で表わせることを見た。しかしながら、このままでは、その計算を実現するためには、加算には加算の計算回路が、乗算には乗算の計算回路が必要になってしまう。桁数ごとに回路を作る必要はなくなったが、計算ごとに回路は必要である。

この問題を解決したのが「プログラムもデータである」という考え方の発見である。



プログラム解釈実行器の概念図、実行したプログラム解釈実行器の概念図。実行したいプログラム P と P に与えるデータを D の二つを共にデータとして受け取り、プログラム P のデータ D に対する実行を行なうのが解釈実行器である。現在のコンピュータの基本構造である。

図 5 プログラム解釈実行器

プログラムそれ自身も文字列であるから 2 進列で表現できる。その 2 進列を処理するのがプログラム解釈実行器と呼ばれているものである。プログラミングの分野では、あるプログラミング言語で書かれたプログラムを解釈実行するプログラムを「インタプリタ」というが、プログラム解釈実行器はその原型といえるだろう。

このプログラム解釈実行器を回路化しておけば、どのような計算も、それをプログラム化して解釈実行器にデータとして与えることにより実現することができる。概念的には、これが現在のコンピュータという電子回路なのである。

例6：単純なプログラムの例

加算と乗算の計算を、計算化の原理にある (1)、(2) 操作のみで表わしてみよう。

$c \leftarrow a$
以下を $b = 0$ になるまで繰り返す

$$\left(\begin{array}{l} c \leftarrow c + 1 \\ b \leftarrow b - 1 \end{array} \right)$$

図 $c \leftarrow a + b$ の計算

$z \leftarrow 0$
以下を $x = 0$ になるまで繰り返す

$$\left(\begin{array}{l} b \leftarrow y \\ \text{以下を } b = 0 \text{ になるまで繰り返す} \\ \left(\begin{array}{l} z \leftarrow z + 1 \\ b \leftarrow b - 1 \end{array} \right) \\ x \leftarrow x - 1 \end{array} \right)$$

図 $z \leftarrow x \times y$ の計算

直感的には左は $c = a$ を初期値として、+1 を b 回行っている。それにより $c \leftarrow a + b$ が実現できる。一方、右の大きな繰り返しの中の小さな繰り返しの部分は左の加算と同じ。つまり、具体的には $z \leftarrow z + y$ の計算である。したがって、全体では $z \leftarrow \underbrace{y+y+\cdots+y}_{x \text{ 個}}$ を計算することで $x \times y$ を実現しているのである。

プログラム自身もデータとして与えることのできる現代型のコンピュータ（プログラム内蔵方式コンピュータ）の本格的なものはイギリスで1949年に始動したEDSAC（Electronic Delay Storage Automatic Calculatorの略）である。フォン・ノイマンのEDVACの提案に触発されて作られた。

このEDSACの前にも電子計算機の一号機ともいえるMark IやENIACが開発されていたが、プログラムをデータと同様に扱うことにより、計算のプログラム化を本格的に実現するコンピュータはEDSACが最初である。この成功により、プログラムを使って計算を設計する、といった考え方が進展し、コンピュータの爆発的な発展が可能となったのである。

用語解説3：フォン・ノイマン

J. フォン・ノイマン（1903～1957）は量子力学の基礎理論のための数学を築いた研究などで有名な数学者であるが、プログラム内蔵方式のコンピュータの創始者としても有名である。

実際、現在のプログラム内蔵型のコンピュータはフォン・ノイマン型コンピュータとも呼ばれている。これはフォン・ノイマンが計算機EDVAC設計レポートで、その考え方を提唱したからである。ただし、最初の発想者は諸説あり、EDVAC設計プロジェクトの中で出てきた考え方、といった方が正しいようである。またチューリングの研究（次節参照）の影響を強く受けた、という説もある。

2.3 計算を見極める研究

ここまで情報を扱う科学技術の二つの重要な基本原理 —デジタル化と計算化— について述べてきた。

この基本原理の創出には「計算」を見つめる研究が重要であった。一方、コンピュータが登場してからは、アルゴリズムの設計やプログラミング技術の開発、さらにはスーパーコンピュータの実践的活用まで、幅広く深い研究が「計算とは何か」を明らかにしてきた。本節では、そうした研究の一端を紹介していく。

2.3.1 計算可能性の研究：「計算」という概念の明確化

デジタル化と計算化の基本原理のうち、とくに計算化の確立に必要な重要な概念の数々は、20世紀の初頭から始まった「計算とは何か」という問いに対する研究の中から生まれてきた。

「計算」を厳密に規定する研究は1930年代に現われ急速に深まった。この研究は、その30年前に出されたヒルベルトの第10問題（用語解説参照）の影響も少なくなかっただろう。ヒルベルトの言う「有限的な方法」が何かを規定しなければ、そのような方法がないという否定的な結果が導けないからである。

1933年～1936年にかけて「計算」に関する重要な三つの概念が相次いで示された。A. チャーチが発表したラムダ計算、それとは独立に、A. チューリングが提案したチューリング機械という計算モデル、そしてその直後に、K.ゲーデルと S. クリーニにより定式化された帰納的関数。

これら見た目のまったく違う概念が、すべて同じ「計算可能性」を定義していたのである。

こうした同値性の結果などから、「計算可能である」という概念を「チューリング機械で計算できる」と定義しよう、という提案がなされた。この提案は現在では、チャーチ-チューリングの提唱またはチャーチの提唱と呼ばれている。我々の計算化の原理も本質的にはこれと同等の考え方である。

この提唱はあくまで考え方であり、数学的に証明することはできない。しかし、現在までに原理的にこの提唱の範囲を超えた「計算」という概念は見つかっておらず、定義としてほぼ定着している。

この計算可能性の研究の最も重要な結果は、計算不可能な問題の発見である。正確に定義できる（ある意味で単純な）関数の中に、計算できないものがある、ということが証明されたのである。その証明のために、チューリングは万能チューリング機械という概念を用いた。これは現代でいうインタプリタである。インタプリタという強力な道具が使えることが、実は計算不可能な問題がある、という計算の限界にもつながっていたのである。

このインタプリタの発見に代表されるように、これまでに紹介した情報科学技術の基本的な考え方の元が、この初期の「計算を定義しよう」という研究の中から生み出されてきたのである。

2.3.2 真の計算可能性の研究：アルゴリズムとプログラミング

計算とは何かという追求は、コンピュータが登場してから数学の研究から大きく踏み出し、本格的に、そして多様な広がりを持つようになっていった。その中でもとくに、計算の設計(アルゴリズム)と計算の表現(プログラミング)に関する

用語解説4：ヒルベルトの第10問題

D. ヒルベルト（1862～1943）は、1900の国際数学会議において20世紀の数学において重要となる問題として23題の問題を指示した。そのうちの10番目の問題「ディオファントス方程式の可解性の決定問題」がヒルベルトの第10問題である。

これは簡単にいえば、不定方程式が整数解を持つか否かを判定する一般的で「有限的な方法」を示せ、という課題である。ただし、この時点では「有限的な方法」とは何かは明確にされていなかった。これは「計算」とは何かを追求するその後の一連の研究で明確になってきたのである。

なお、この第10問題は、1970年にロシアの数学者 J.V. マチャーセビッチによって否定的に解決された。つまり、ヒルベルトが求めた方法は存在しないことが証明されたのである。

研究に焦点をあててみよう。

情報を扱う科学技術の基本は、様々な仕事をコンピュータで行うこと、そのために仕事を、その対象も含め、コンピュータ上に載せること、つまりデジタル化と計算化である。そこには現実の(場合によっては未来の)コンピュータがあり、情報科学技術は、その上での実現をつねに意識したものになっている。

このように実現を意識したときに直面する最大の問題は「量と複雑さ」である。大きさかもしれないが、コンピュータが登場してはじめて「量と複雑さ」が人類にとって身近なものになったといってもよいだろう。情報を扱う科学技術の研究は、この「量と複雑さ」にいかに関わり深いかの研究なのである。ここではそうして研究の中で「計算」にとくに関わりの深いアルゴリズムとプログラミングに関する研究の一端を紹介する。

アルゴリズム (algorithm) とは、計算を行う手順のことである。計算化の原理でも述べたように、あらゆる計算は単純な演算を組み合わせることで構成できる。構成要素が単純なだけに、その組み合わせ方は様々で、非常に自由度がある。同じ仕事を処理させるのでも、そのやり方は千差万別である。その「やり方」すなわち処理の手順がアルゴリズムである。

解説4：チューリング機械

A. チューリングが計算の抽象モデルとして提案した仮想機械。最も基本的なものは、右図のように、マス目に区切られた無限に長いテープの上を、テープヘッドが走査するもの。ヘッドが動き、ヘッドの位置するマス上の記号を読みとったり、マス上に記号を書きかえたりする。ヘッドが動くと、この読み書きを行い、テープヘッド中の有限種類の状態も変化させる。これを有限状態が停止状態になるまで続ける。

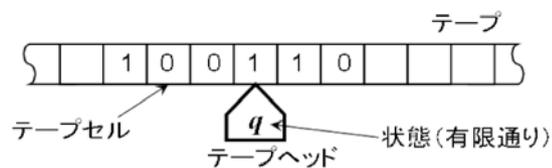


図 チューリング機械

計算と見る場合には、テープに最初に与えられた 0、1 の 2 進列が入力であり、それに対して停止状態時にテープ上に得られる 2 進列が計算結果の出力である。

チューリング機械は、我々が議論してきた計算の基本要素を単純化し抽象化したものになっている。計算の過程を表わす「状態」は、テープの内容、テープヘッドの位置、そしてテープヘッド内の状態であり、それらを変えることが「基本動作」である。その基本動作を停止状態になるまで「繰り返す」ことで計算を行なっているのである。この三要素から、我々の計算化の原理と同様、すべての計算はチューリング機械で実現できるのである。

このチューリング機械の提案とともに、チューリングが発見した重要な事実は、すべてのチューリング機械を模倣する汎用のチューリング機械 (万能チューリング機械) が作れることであった。現代の言葉でいえばインタプリタの存在性を証明したのである。

計算の実現を意識したとき、まず重要になってくるのが「資源」である。どのくらいの計算時間が使えるのか、どのくらいのメモリ領域(記憶領域)が計算に使えるのか、といった点である。同じ仕事や計算をするのでも、アルゴリズムの良し悪しで、この資源の使われ方(コスト)に天と地の差が出てくる。適切なアルゴリズムを使っていれば数分ですむ仕事が、効率の悪いアルゴリズムを使ったために数日かかってしまう場合もある。あるいは、単純には数年かかるといわれた計算も、非常に巧妙なアルゴリズムが発見されたことで数時間で済む場合もある。

アルゴリズムの効率の良し悪しを議論する場合、計算量 (computational complexity) という尺度を用いることが多い。これは問題の規模の増加に対して、そのアルゴリズムの実行に必要な計算資源(たとえば計算時間)が、データ量 n の増加に伴って、どのように増えるかを表わす関数である。

計算量を表わす場合には通常、「時間計算量が $O(n^2)$ のアルゴリズム」という言い方のように、比例定数の部分は省略した漸近的表現が用いられている。

これは比例定数まで細かく評価するのが難しい、あるいは、そこまで細かい理論的な評価の必要性が低い、という理由からである。

もちろん、実用上は比例定数の部分も重要である。一般的には計算量が小さいアルゴリズムの方が効率的だが、実際のデータ量では必ずしもそうならない場合がある。たとえば時間計算量が $O(n^2)$ のアルゴリズム A と、 $O(n^4)$ の B があつたとする。オーダーでは B の方が速いが、実際には A の方がはるかに早い場合もある。A の方が単純で比例定数が小さく、実際の問題の規模では B より速い場合などである。

このように定数係数については注意が必要だが、それでも漸近的計算量の低いアルゴリズムの方が有利であることが多い。実際の数値例で考えてみよう。

たとえば、アルゴリズム B の実際の計算時間が $4n^4$ 秒だったのに対し、もう一方のアルゴリズム A が $900n^2$ 秒かかったとしよう。このとき、もし問題のデータ

用語解説5：漸近的表現、オーダー記法

関数の値の増減の速さの程度を大づかみに表現する方法。一つか二つのパラメータに対する変化を記述し、比例定数などは省略する方法である。一般的に用いられているのは $O(n^2)$ (n の 2 乗のオーダー) という言い方である。現在考えている関数が n の 2 乗に比例して増加する、という意味である。

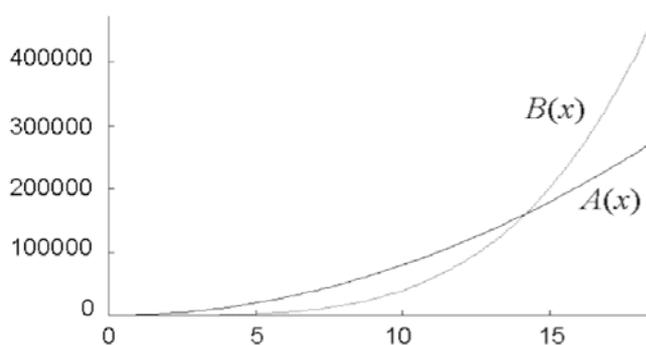


図6 増加率の違い A: $O(n^2)$ と B: $O(n^4)$

量が $n = 10$ だったとすると、A は 90000 秒 (25 時間)、B は 40000 秒 (11 時間) である。しかし、データ量が $n = 20$ になると、A の方が圧倒的に有利である。

さらに、1 時間で処理できるデータ量を比べてみる。A では $900n^2 \leq 3600$ 秒という式から $n = 2$ 以下となるが、B では $4n^4 \leq 3600$ 秒から、 $n = 5$ 以下である。B の方が少し大きなデータ量まで扱える。しかし、たとえばコンピュータの速度が 1 万倍になったら、どうだろう。A は $900n^2 \div 10000 \leq 3600$ 秒から、 $n \leq 200$ となるが、一方、B は $4n^4 \div 10000 \leq 3600$ 秒から、 $n \leq 55$ 程度である。A が 100 倍大きなデータを扱えるようになるのに対し、B は 10 倍強である。

つまり、B は、データ量が大きくなると早く手に負えなくなるだけでなく、コンピュータの性能向上の効果が小さくしか現れないのである。

一般に漸近的計算量の低いアルゴリズムは、コンピュータの速度の向上の恩恵を受けやすく、問題の規模の増加にも安定的である。「アルゴリズムの改良などより、コンピュータを速くした方が手っ取り早い」と豪語する人もいるが、実は、その逆で、コンピュータの性能が上がれば上がるほど、よいアルゴリズムが重要になってくるのである。

大ざっぱに言えば次のような考え方が重要だろう。

効率化の極意

アルゴリズムで本質的な速度向上をはかり、
プログラミングで実質的なスピードアップをし、
そしてコンピュータの性能で最後の詰めをする。

解説5：実際の計算の時間は？

実際の計算時間はコンピュータの性能によって大きくことなる。その性能の測り方は一通りではないが、たとえば、小数演算を 1 秒間に何回行なえるかを測る FLOPS という単位がある。たとえば、1 ギガ FLOPS は、1 秒間に 10 億回の計算である。

右の表は、様々な計算装置の最適な環境の下での FLOPS 値の概算を示したものである。最近のゲーム器は、特に小数演算に特化して良い性能を出すように設計されているのでこの数値が高い。ただし、これはあくまで一つの指標にすぎず、この数値が高い＝実際の計算性能がよい、とは限らない。

ゲーム器の演算装置	数十ギガ FLOPS
パソコン	
初期のスパコン	数百メガ FLOPS
最速スパコン	数百テラ FLOPS
次世代スパコン	1 ペタ FLOPS は
初代コンピュータ	数百 FLOPS
	(2007 年)

1 メガ=100 万、1 ギガ=10 億
1 テラ=1 兆、1 ペタ=1000 兆

様々な計算装置の演算能力

1 ギガの計算とは？ (大ざっぱな見積り)

- 1000×1000 の行列同士の掛け算
- 100 億 (10 桁) 程度の数の素因数分解
- 0 と 1 の数字が 30 個ある列をすべて見る

効率の良いアルゴリズム開発の研究は、いまやそれ自身が数理科学の重要な分野になるまでに発展した。しかし、残念ながら、どんな問題にも効率のよいアルゴリズムが作れるとは限らない。計算不可能な問題があるのと同様、たとえば $O(n^2)$ 時間では絶対解けない問題、 $O(n^{10})$ 時間でも無理な問題等々が存在することが証明されている。

その一方で、効率よく解けるか否かがよくわかっていない問題も多い。その代表例が NP 問題（正確には NP 完全問題）と呼ばれる一群の問題である。NP 問題は、実に様々な分野で見つかっており、コンピュータの応用上重要な問題も多い。多くの研究者は「NP 完全問題を一般的に効率よく解く方法はない」と信じている。これを $P \neq NP$ 予想という。

NP 完全問題を一般的に効率よく解く方法はない、と予想されているからといって手をこまねているわけにはいかない。実際には解かなければならない問題も多い。一般的とまではいかななくても部分的に解けてもよいし、よい近似解が得られてもよい。あるいは、時間計算量が漸近的には大きくても、定数係数が非常に小さくて、ある程度の規模の問題までならば実用に耐えうるアルゴリズムでもよいかもしれない。様々なアルゴリズム的な試みがなされているのである。

解説 6 : NP 問題と NP 完全問題

世の中には、答えを言われれば「なるほど」とすぐ納得できる問題が少なくない。大ざっぱに言えば、NP 問題は、そうした性質を持つ計算の問題の総称である。

例をあげよう。「この数を同じ桁数の積で表わせ」という問題がその一例である。たとえば 221 が問題例で、その解は 13×17 である。この答えが正しいかを確かめるのは簡単だろう。実際に積が 221 になることを確かめればよい。

あるいは、与えられた 100 都市を、150 日間で訪問して回るような旅程が組めるか？という問題もその一例である（これは巡回セールスマン問題と呼ばれている）。都市の巡り方を言われれば、それが目標の旅程であるかをチェックするのは難しくない。

パズルのような問題も多い。たとえば、三彩色問題である。これは、与えられたグラフに対し、その各頂点に色を塗りたい、ただし、辺で結ばれた頂点同士が同じ色にならないように、しかも三色で塗り分ける方法を求めよ、という問題である。これも答えを言われれば確かめるのは簡単である。

答えを言われれば簡単に確かめられるとしても、それを探すのは容易とは限らない。たとえば、6 頂点のグラフを 3 色で塗り分ける方法を全部試してみると、 $3^6 = 729$ 通りもある。もし頂点数が 50 になったら大変だ。

もちろん、うまく解を見つける方法があるかもしれない。けれども NP 問題の中には、うまく解を見つける効率のよい一般的な方法が見つからないものも沢山ある。その中でも最も難しいということが示されている問題群を NP 完全問題という。実は、巡回セールスマン問題、三彩色問題は NP 完全問題の一つである。

次に、まったく異なる「量と複雑さ」に対処する話に目を向けよう。アルゴリズムは計算の設計に関する話だったが、計算の表現に関わる話題である。

計算化の原理でも述べているように、すべての計算は単純な演算から組み立てられる。しかし、このように原理的に可能であってもその大きさ、複雑さは膨大なものになる。基本演算の機能を大幅に強化したとしても、現在の少々大きなシステムとなると、プログラムで数万行になることも少なくない。

このように大規模になってくると、「原理的には表わせる」と言っても意味がなく、実際にどのように計算を表わすか、という方法論が重要になってくる。その方法論の提案とその直接の影響・成果が、今日のプログラミング言語である。コンピュータの出現直後から、より書きやすい、よりわかりやすい計算の表し方を目指して、様々なプログラミング言語が提案されてきた。そこには、サブルーチン、スタック、再帰、ポインタなど、計算をうまく表現するために重要な工夫が取り入れられていった。

ただし、これらの工夫だけでは、プログラムの急速な大規模化・複雑化に対処しきれない。新たな方法論が必要になってきたのである。人間が考えやすく、間違いをしにくく、また修正や改訂なども容易に行えるようなプログラミングの方法論として、関数型プログラミングやオブジェクト指向プログラミングという考え方が出てきた。とくにオブジェクト指向は、その考え方に沿ってプログラムを作成するためのプログラミング言語 C++ や Java などが提供され、今や大規模なプログラミングには欠かせない方法論になっている。

いくら方法論がよくても人間ができることには限りがある。そこで人間のプログラム作成を支援するための様々なプログラミングツール（プログラムを作成する道具としてのソフト）が開発されている。プログラムのタイプ入力を助けるために登場したエディタであったが、それが発展し、今やプログラミング方法論を取り入れ、それに自然に従ってプログラムを作成できるようなプログラム統合開発環境を提供しているシステムもある。それにとまって、最初からアルゴリズムを設計して作るのではなく、既存のプログラム部品を組み合わせるプログラムを作成する場合も多くなってきた。

一方、できあがったプログラムの正しさを保証する手法についても多くの研究がなされてきた。プログラムが所望の動作をしないというのは、単にプログラム作成上のミスだけでなく、そもそも設計段階で、システム作成を依頼した側と作り手との間に誤解がある場合も多い。このような齟齬が生じないためのプログラムやシステム設計手法、そしてシステム作成の各段階での動作検証手法などの研究が行われている。

さらにはテストによる検証だけでなく、システムが正しく動くことを検証（証明）したい、という場合もある。医療や交通、あるいは原発などの非常に重要な

システムや多数のパラメータが介在する複雑な情報セキュリティシステムなどでは、正しく動くことが「証明」できれば、より安心である。これまでは、計算量の問題などで、機械的な証明を実際に行うことは考えられていなかった。しかし、アルゴリズムの進歩、そして適用範囲の適切な選択により、一部の応用では証明も十分可能な段階になりつつある。

2.3.3 計算世界観：計算を中心とした新たな観点

コンピュータに載せるためのデジタル化・計算化の研究の進展により、情報を扱う様々な科学技術が発見され、また開発されてきたが、それらはコンピュータのためだけでなく、一般的な科学分析の手法として、他の分野での研究でも使われ、その分野の研究に影響を与えている場合もある。

コンピュータの道具としての影響は当然のことながら、コンピュータを利用して情報を扱うために研究されてきた科学技術自身が「科学の手法」として他分野へ影響するようになるにつれ、情報科学技術をコンピュータのためだけでなく、科学分析の手法として用いるという考え方も生まれてきている。単にコンピュータを道具として使うためのデジタル化・計算化だけでなく、様々な現象や事柄を「計算」を通してみよう、という新たな考え方の芽生えである。こうしたアプローチを計算世界観（computational view、computational thinking）と呼ぶことにしよう。

「すべては計算で表わせる」というのは偏った考え方だろう。危険ですらある。けれども、人類が記号や式、あるいは科学的な文で議論できる多くの事柄は、計算で表わせると考えてもよいだろう。そこで逆に、計算を通して物事を分析するのが計算世界観である。

計算を中心に見ることで、計算に適した判断ができる場合がある。様々な現象をモデル化する場合、通常は完璧なモデルを作ることは期待できない。何らかの妥協をしなければならないのである。計算を中心に見ることで、計算に有利な妥協点を見出せる可能性がある。新たな切り口を見出せるのである。

計算という見方で見ることで新しい発見が得られる場合も少なくない。たとえば、NP 完全問題の一つである三彩色問題の高速解決法が見つければ、素因数分解も高速にできる。ということが証明されている。一見して何の関連もない二つの問題が、計算を通して見ると密接に関係しているのである。これはかなり専門的な事例ではあるが、一般にも、計算という見方で新しい発想や概念を得られる場合があり、計算という切り口は、これからの科学のアプローチとして重要になってくるであろう。

例7: 情報を扱う科学技術が他分野へも影響している例

プログラミング言語の解析手法の言語研究への影響

人が理解しやすいプログラミング言語が数々と開発されたが、そのようなプログラミング言語で書かれたプログラムを、機械にもわかる命令列に変換する（つまり翻訳する）必要がある。プログラムを正確かつ高速に（機械語に）翻訳するため、文（プログラム）の構文解析技術の研究が非常に進展した。

チョムスキーらは文法を数理科学的に議論する枠組みを提案した。そもそも言語学の体系化の試みのために提案されたものだが、それがプログラムの構文構造の体系化に適用され、チョムスキーの枠組みのもとで、構文解析技術が大きく進展した。

その応用として、今度は、我々が通常使う言語（これをプログラミング言語と対比させて自然言語という）の解析へも用いられるようになったのである。こうした研究の展開は自然言語処理という新たな分野を生み出した。その中では、たとえば日本語から英語への機械翻訳や、辞書の検索技術、そして、様々な検索を容易にする辞書の構成法などの研究が進んだ。

こうした研究の中で見出された様々な科学技術は、本来の言語理論にも大きな影響を与えるようになってきた。それらは単にコンピュータを利用するための技術にとどまらず、言葉の意味や用法について、新たな知見を得るための「科学の手法」として注目を集めるようになってきたのである。

例8: 計算を通して見ることが新たな観点を導き出した例

ランダムな明確な定義

確率論の基礎に貢献した A. コルモゴロフは、ランダムという概念を明確に定義しようと試みた。その頃ちょうど明確になりつつある「計算」という概念を用いることを考えたのである。彼のこのアプローチは、コルモゴロフとその関係者により達成され、ランダムという概念は計算という見方のもとではじめて明確な定義を得たのである。

公開鍵暗号系

公開鍵暗号系は、通信理論で研究されてきた暗号の分野に、「計算」という見方を持ち込んだために生まれてきた。

暗号通信を盗聴する者も、結局は限られた計算資源の元に暗号解読しなければならない。逆に言えば、限られた計算資源の元では解読できない暗号を考えればよいのである。この考え方が公開鍵暗号を生み出したのである（公開鍵暗号についての説明は第 3 章参照）。

私たちはユーザーとしての様々な体験から、専門家の知の体系とは異なる「便利さの体系」（ユーザーの視点から情報技術によって可能になったことを体系化したもの）を学んでいる。情報学リテラシーとして求められているのは、そのような便利さと情報学とのかかわりを知り、より豊かな生活を送ることができるようになることであろう。本部会では、情報学リテラシーを、原理、仕組み、意義、影響の四層で捉える観点で検討を進めてきた。情報学リテラシーの学び方をこの観点から見たとき、原理・仕組みと私たちの体験とをいかにつなげるか、という論点は重要である。

何かを学ぶときに、題材が学習者にとって身近であることは、重要な条件の一つである。ところが、専門家が考える身近さと、学習者が考える身近さとは、往々にして一致していない場合が多い。以前、中学校の理科教師から、中学生が慣性の法則を実感できるのは、スペースシャトルや人工衛星の映像を見た時であり、目の前のだるま落としては十分ではない、と指摘を受けたことがある。また、身近ではない題材で法則の学習を続けていると、生徒たちは、自分たちの身近な世界でその法則が成立しているという感覚を抱けずに、「知っているが信用しない」知識として理科学習用の引き出しに入れ、鍵をかけてしまうという。先に述べた「便利さの体系」は、私たちの生活に十分に身近であるが、それは、情報学の成果が応用され、実現された結果であって、情報学が追究してきた仕組みや原理とのギャップは大きい。便利さの体系と「つなげる」ことなく、原理・仕組みを学ぶのだとしたら、学習者は、それを情報学リテラシー用の引き出しに入れ、鍵をかけてしまうだろう。

便利さの体系と原理・仕組みを「つなぐ」ものは、たとえば、次のようなものではないだろうか。

- ・ 情報学研究者の努力：研究者や技術者の実際の活動を知ることにより、便利さは人間の知恵と努力の成果であることがわかる。
- ・ 情報技術開発の基本的な方法であるプログラミングの存在：身近な情報技術の産物が、「作られた」ものであることがわかる。

情報学は、単純な原理から様々な仕組みを実現、発展させてきたが、私たちの便利さを享受する体験は、その過程と切り離されている。そのため、情報学に携わる人々の知恵や努力、方法そのものが原理・仕組みと体験を「つなぎ」、信用できる情報学リテラシーを学ぶために必要と考えるのである。

第3章 情報を扱う科学技術の仕組み

情報を扱う科学技術は、分野・観点・レベルに応じて、様々な仕組みがある。しかしながら、それらの仕組みにおいてもいくつかの共通した考え方がある。共通した考え方の代表的なものとして、抽象化と仮想化があげられる。本章では、まずコンピュータの基本構造を説明した後に、抽象化と仮想化という二つの観点から、情報を扱う科学技術の仕組みを紹介する。そして、いくつかの重要な応用技術の仕組みについて解説する。最後に、コンピュータの発祥・発展を支えた電子回路技術について補足する。

3.1 コンピュータの基本構造：電子機器として

コンピュータは集積回路という部品を中心に作られた電子機器である。現在使用されているコンピュータは基本的には次の要素から構成されている。

- ▶ **プロセッサ**：数値の計算を行ったり、条件を判定したり、入力と出力（以下では入出力）に指示する。CPU もしくは中央処理装置（central processor unit）とも呼ぶ。そして、プロセッサは、大まかには次の二つに分かれる。
 - ・ **データパス**：演算処理（計算やデータの加工など）を行なう。体に例えると、「手や足」である。一時的にデータを格納するレジスタや演算装置などからなる。
 - ・ **制御装置**：何を行うべきかをデータパスや入出力に指示する。体に例えると「神経系」である。
- ▶ **メモリ**：実行中のプログラムと、プログラムが必要とするデータを格納する場所である。記憶装置とも呼ばれる。

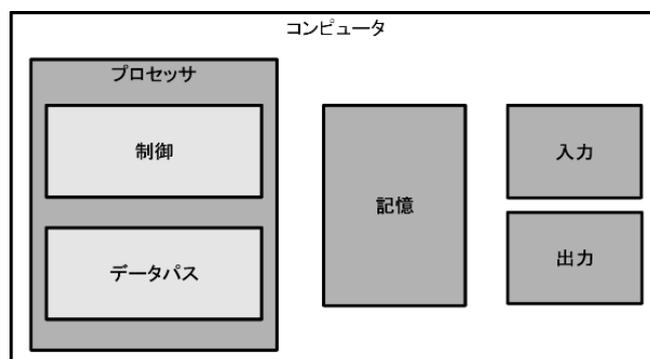


図7 コンピュータの基本構成の概念図

- ▶ **入出力装置**：I/O とも呼ぶ。キーボードやマウスなど、コンピュータにデータを入力するものや、ディスプレイのようにコンピュータからデータを出力するものがこれにあたる。ネットワークのように、入力と出力の両方の機能を持つものもある。

この基本構造における重要な点の一つに、コンピュータの動作を決めるプログラムがメモリに格納されるということがあげられる。これにより、プログラムを変更することによりコンピュータの動作を変えることができる。パソコンにゲームプログラムを入れればゲームマシンとなり、ワープロプログラムを入れればワープロとなるのは、この特徴によるものである。メモリにプログラムを搭載するというコンピュータの基本構造は、プログラム内蔵方式と呼ばれる。メモリに格納されたプログラムは、データとしてプログラムによって変更の対象になるという点も重要である。コンピュータを停止することなしに、別のプログラムを投入することができるのは、プログラム内蔵方式によるものである。メモリに格納されたプログラムをその他のデータと区別することなく扱うことができるような構造はフォン・ノイマン型アーキテクチャと呼ばれる。これに対して、データとプログラムとを区別して扱う構造をハーバードアーキテクチャと呼ぶ。

最初の現代的なコンピュータである EDSAC が開発されて以来、コンピュータは飛躍的な進歩をとげたが、このフォン・ノイマン型アーキテクチャは、そうした様々なコンピュータに共通した、ほぼ不変の構造となっている。

コンピュータは電子工学的に見れば（たとえ小さなパソコンといえども）巨大な電子回路である。たとえばプロセッサを実現するインテル社の Core2Duo の場合、2 億 9100 万個のトランジスタから構成されている。したがって、コンピュータは電子回路技術の発展なしには生まれもしなかったし、また、社会的に影響を与える存在にもならなかっただろう。けれども以下では、そうした電子工学的な側面は省略し、コンピュータならびにコンピュータを中心とした情報システムの情報科学技術的な仕組みについて主に述べる。電子工学的な側面については、最後の 3.7 節で補足する。

3.2 コンピュータの基本構造：ソフトウェアシステムとして

コンピュータという一つのシステムは、ハードウェアとソフトウェアの二つに大別することができる。ハードウェアは元来、「金物」を表す言葉であった。コンピュータの世界では、電子機器としてのコンピュータ本体やそれを構成する部品をハードウェアと呼ぶことが多い。前節は、そのハードウェアとしてのコンピュータの基本構造の説明だった。

コンピュータは、ハードウェアだけでは動かない。そのハードウェア上に処理

を実現するためのソフトウェアが必要である。その実体は多数のプログラムである。そうしたプログラム群の総称がソフトウェアと考えてもよいだろう。本節では、ソフトウェアに関して、その目的により分類しながら説明する。

一般に、「ソフト」と言われるものは、アプリケーションソフトウェアと呼ばれるものであり、特定の用途をもったものである。たとえば、文書の作成に関しては、文書の編集作業を目的とするテキストエディタ、そして、文字の装飾や段落の幅などの文書の体裁を整えることを目的としたテキストフォーマッタ、この二つの機能を兼ね備えたワープロソフトなどがあげられる。

数値データの集計や分析を主な目的とするソフトウェアとして表計算ソフトウェアがある。これは、集計用紙を模した格子状に並んだセルと呼ばれる欄に数値や、数値と数値との関係を数式で記述することにより、数値データの計算をおこなう。現在では、単なる数値データの集計・分析に留まらず、事務における小規模な帳票処理等に幅広く用いられている。

以上の二つ、ならびに後で解説するインターネット利用のための通信・閲覧・検索のソフトウェアが一般的かつ代表的なソフトウェアだろう。その他にも、多項式の展開や因数分解、微分や不定積分などを記号的な処理でおこなう数式処理システムと呼ばれるソフトウェアがある。これは、理工系分野の研究者や学生に広く役立っているソフトウェアの一例である。

これらのアプリケーションソフトウェアは、ハードウェアとしてのコンピュータ上で直接動作するのではなく、オペレーティングシステムと呼ばれるソフトウェアが両者の仲立ちとなる。オペレーティングシステム自体は、文書作成やデータ集計といった特定の用途をもたず、ハードウェアの制御や管理をアプリケーションソフトウェアの代りに一括して行なう。そして、オペレーティングシステムは、個々のアプリケーションソフトウェアに共通して必要となる機能を実現する。このように個々の用途に依存するのではない一般的な目的をもったソフトウェアをシステムソフトウェアと呼ぶ。

システムソフトウェアには、オペレーティングシステムの他、プログラミング言語処理系や、アプリケーションソフトウェアに共通した機能を担うミドルウェアが、あげられる。

ミドルウェアの例としては、データベースを管理するためのソフトウェアである DBMS や Web ページの配信を担当する Web サーバソフトウェアなどがあげられる。ローマ字を平仮名・片仮名・漢字に変換するかな漢字変換システムも、システムソフトウェアの一つであるといえる。

また、ライブラリと呼ばれるソフトウェアの部品の集合体がある。これもシステムソフトウェアの一つであり、たとえば、OpenGL と呼ばれる 3 次元画像の表示のためのライブラリがその典型である。これは、アプリケーションソフトウェア

が画像表示のためのハードウェアを直接操作するよりも、効率的・効果的に 3 次元画像を表示することを可能にしている。

3.3 抽象化の観点から見た情報科学技術の仕組み

抽象化とは、ある性質に着目して物事を単純化して考える手法である。まず、具体的な例として、メモリ（主記憶装置）に見られる抽象化について説明する。

メモリは 0 と 1 の列によって表現された情報を格納する。メモリには番地（アドレス）がふられていて、各番地に対応して一定の長さの 2 進列が格納されている。プロセッサなどの他の装置からの情報の出し入れは、番地を指定することにより行われる。一つの番地に対して、長さ 8 の（8 ビットの）2 進列が格納されている。これを表わしたのが、図 8 である。

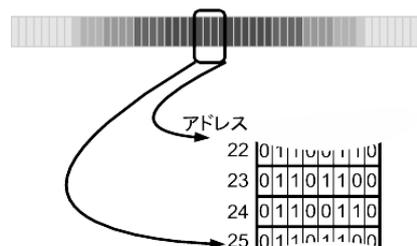


図 8 メモリの概念図

メモリの構造は、実際にはもっと複雑である。たとえば、パソコンのメモリは DRAM と呼ばれる電子部品から構成されている。その概念図を図 9 に示す。

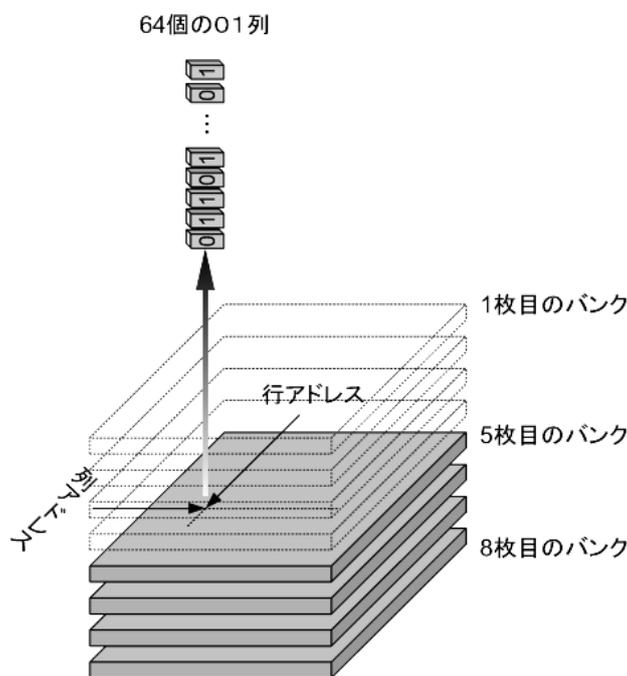


図 9 DRAM の概念図

DRAM は、図 8 に示したような帯状の一次的な構造ではなく、京都や長安の町のように碁盤状の二次元の構造になっている。行アドレスと列アドレスとを与えて、各区画からデータを取り出したり保存したりする。各区画には 64 ビットの 2

進列が格納される。さらに、そのような基盤状の記憶領域（バンク）が複数存在することがある。その場合、一つの区画を指定するには、行アドレスと列アドレスと、そして何枚目のバンクであるのかを指定することになる。

ただ、各区画に格納されるデータの長さは 64 であるとは限らず、32 だったり 16 だったりする。また、バンクの個数、行アドレスの範囲、列アドレスの範囲は、その DRAM の容量や種類によりさまざまな構造、さまざまな規模のものが生産されている。これらの違いは、プロセッサとメモリとの間に、いくつかの仕組みが設けられていて、図 10 のような構造に見えるように抽象化されている。メモリが抽象化されることにより、プロセッサはその違いを意識することなしにメモリを取り扱うことができるのである。

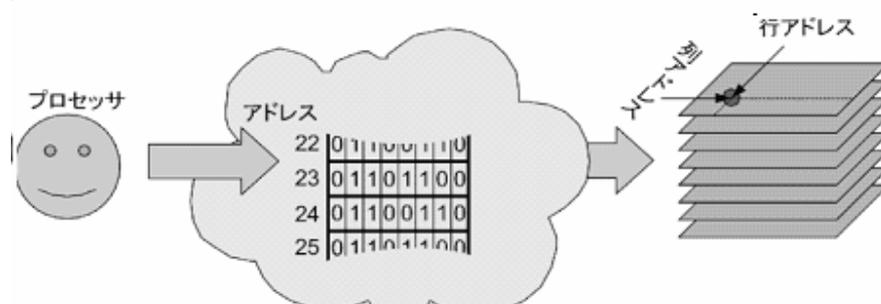


図 10 メモリにおける抽象化

抽象化が行われるのは、メモリの場合に見られるような、ハードウェアとハードウェアの間にとどまらない。ハードウェアとソフトウェアとの間、ソフトウェアとソフトウェアの間など、さまざまな場面で抽象化は行われている。

そのもっとも代表的な例は、オペレーティングシステムである。これは、アプリケーションソフトウェアなどのコンピュータ上で動作する様々なソフトウェアに対して、ハードウェアを抽象化して見せるソフトウェアであるということができる。

コンピュータにおいてプロセッサが同種類のものであっても、ハードウェアが異なることは多い。たとえば、デジタルカメラで撮影した画像をパソコンで処理する場合を思いうかべてほしい。画像ファイルがメモ리카ード（SDメモ리카ードなど）の上にあっても、ハードディスク上にあっても、画像を処理するソフトウェアでは同じように扱うことができる。しかし、ハードウェアとしては、メモ리카ードとハードディスクはまったく異なる。画像処理ソフトウェアがこの二つの異なるハードウェア上にあるファイルを同一に扱うことができるのは、オペレーティングシステムがソフトウェアとの仲立ちになるからである。

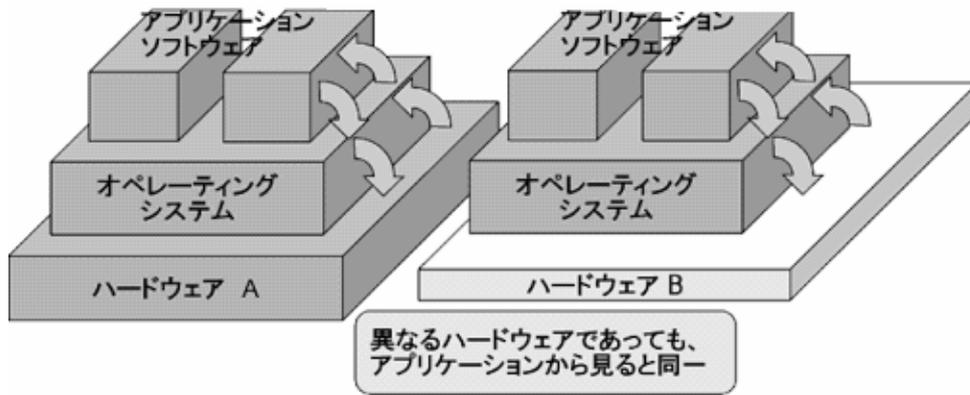


図 11 オペレーティングシステムの概念図

3.4 仮想化の観点から見た情報科学技術の仕組み

仮想化とは、実際に存在しないものをあたかも存在しているかのように見せかけることである。次に、この仮想化という観点から仕組みをながめてみよう。まず、プログラミング言語における仮想化から述べよう。

3.4.1 プログラミング言語の仮想化

プログラムとは、コンピュータがどのように動作するのかということを記述したものである。プログラムとは一定の規則にのっとって記述される。その規則をプログラミング言語と呼ぶ。プログラミング言語は、プログラムを記述するために作られた人工言語であり、文法は、コンピュータが解釈して実行できるように作られている。プログラミング言語には様々なものが存在するが、その中でもっとも基本的なプログラミング言語が機械語である。

機械語は、コンピュータ（正確に言うとうと CPU）が直接解釈して実行することができる言語である。機械語では、ある長さの 0 と 1 の列が CPU の基本命令に対応づけられている。機械語の基本命令自体は単純であるが、単純であるがゆえに機械語プログラムは可読性に欠け、人間が理解するのには困難が伴う。その問題を解消するために使われているのが、高水準プログラミング言語（以下では単に高水準言語）と呼ばれるプログラミング言語である。この高水準プログラミング言語で仮想化の思想が用いられているのである。

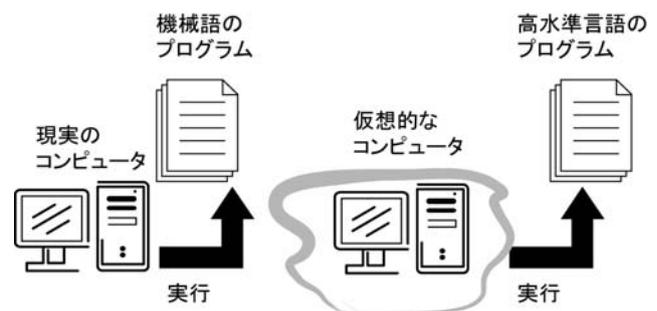


図 12 機械語と高水準言語

機械語プログラムは、具体的な演算装置 CPU が想定されているのに対して、高

水準プログラミング言語においては、高度な仮想的なコンピュータが想定されている。

この「高度な仮想的なコンピュータ」は計算モデルと呼ばれ、計算モデルがどのような処理を行うのかを定義したものをプログラミング言語意味論と呼ぶ。そしてプログラミング言語意味論に基づいて、プログラミング言語処理系（以下では単に言語処理系）といわれるソフトウェアが作られる。言語処理系を用いて、高水準プログラミング言語のプログラムは実行されることになる。（プログラミング言語意味論を研究する分野も「プログラミング言語意味論」と呼ばれるので注意が必要である。この分野では、プログラミング言語意味論を数学的に記述し、その構造を研究する。言語処理系が想定する意味論が数学的に記述されたものであることもあるが、そうでないことが多い。）

高水準言語の長所は以下のようにまとめられる。

- ・各種のプロセッサや入出力装置とは独立に作られている。
- ・具体的なコンピュータに依存しない計算モデルが想定されていて、プログラムは計算モデル上で動作すると考えられている。ここでいう計算モデルとは「仮想的なコンピュータ」である。
- ・計算モデルは、多くの場合、ある特定の種類のソフトウェアが記述しやすいように作られている。低水準言語と比べて、より抽象的なものとなっている。

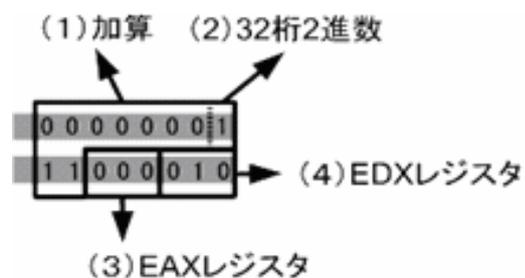
これらの長所が得られたのは高水準プログラムが「仮想的なコンピュータで実行される」という考え方のためである。「仮想的なコンピュータ」の上の言語を考えたお陰である。

例9：機械語命令

機械語の命令の具体例一つを紹介する。下図は、インテル社の x86 シリーズのプロセッサ（Pentium、Core2Duo など）の加算命令である。このプロセッサは、0 と 1 が 16 個からなる列（図では二段で表示）が 1 つの命令を表わす決まりになっている。

0000000111000010

この 2 進列が表わしている命令
一時記憶領域（EAX レジスタ）
に格納されている整数と別の一時
記憶領域（EDX レジスタ）に格納
されている整数を足して、その結
果を EDX レジスタに入れなさい。



高水準言語の技術上の要点は、具体的なコンピュータにとらわれない「仮想的なコンピュータ」すなわち、計算モデルを想定し、それに基づくプログラミング言語を設計しているところにある。目標とする計算の記述方法の特性にあった計算モデルを用いることで、適切なプログラミング言語を設計することにより、プログラム作成効率が飛躍的に向上するのである。

計算モデルと高水準言語は、情報科学技術における仮想化の典型例であるといえる。コンピュータを計算モデルに仮想化することにより、プログラム作成者の思考を実際のコンピュータから解放し、プログラムを作成しやすくしているのである。

3.4.2 オペレーティングシステムと仮想化

オペレーティングシステムにおける抽象化については前に述べたが、仮想化もオペレーティングシステムにおいて重要な役割を果たしている。オペレーティン

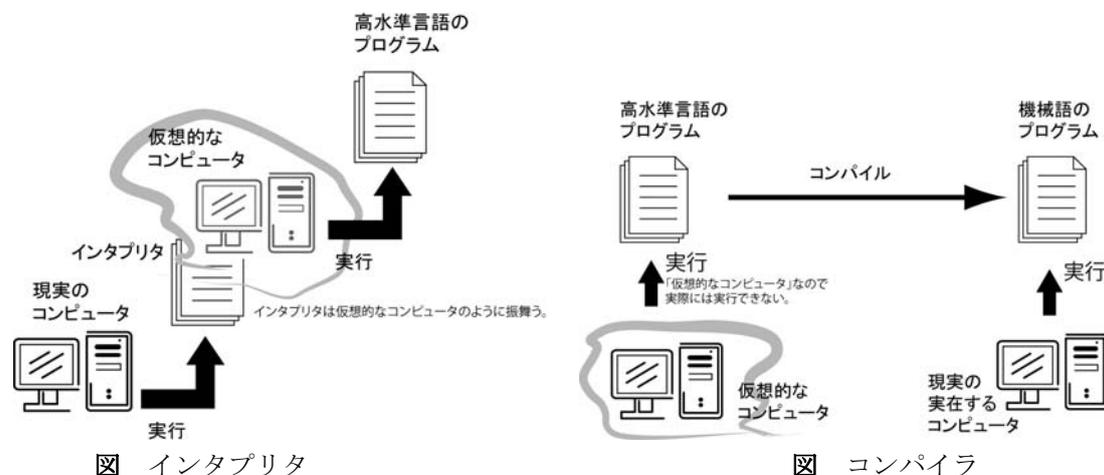
解説7：言語処理系：高水準プログラミング言語が使えるわけ

高水準プログラムが使えるのは、コンパイラとインタプリタと呼ばれる言語処理系のお陰である。

- ・インタプリタは高水準言語のプログラムを解釈し実行するソフトウェアである。
- ・コンパイラは高水準言語を機械語プログラムに変換するソフトウェアである（変換された機械語プログラムはプロセッサで実行される）。

インタプリタは、高水準言語を解釈・実行する計算モデルを表現したソフトウェアである。別のいいかたをすると、コンピュータはインタプリタにより高水準言語を機械語とするようなコンピュータに「化ける」のである。

一方、コンパイラにおいては、高水準言語が想定する計算モデルの位置付けは間接的なものとなる。高水準言語が想定する計算モデルはコンパイラという変換を通してコンピュータと対応づけられている。



グシステムは、その上で動作するアプリケーションシステムに対して、抽象化され、かつ、仮想化されたコンピュータの姿を見せている。

仮想化の典型的な仕組みが、マルチタスキングである。これは複数のプログラムの処理を同時に行うもので、CPU が一つしかなかったとしても、一つの CPU を十分短い時間感覚で切り替えて複数のプログラムが同時に動作しているように見せかける仕組みである。CPU の処理速度と比例して、CPU 以外の装置の処理速度やコンピュータを利用する人間の処理速度が圧倒的に遅いため、CPU を十分短い時間で切り替えればあたかも複数の CPU が存在するように見せかけることができるのである。

他にもオペレーティングシステムでは、仮想記憶という実際には存在しない量のメモリをあたかも存在するように見せかける仕組みがある。

他の例としては仮想機械というソフトウェアがある。オペレーティングシステムが提供する仮想化は、抽象化の仕組みと組み合わせっており、オペレーティングシステムで動作するプログラムから見ると、プログラムが記述しやすいような理想的な環境がコンピュータのイメージとして提供されている。それに対して、仮想機械ではもっと具体的なコンピュータのイメージが提供されている。実際に存在するコンピュータ、たとえば、インテル x86 シリーズの CPU を搭載したパソコンを仮想的に実現した仮想機械も存在する。言い方を変えると、これはコンピュータそのものを仮想的に実現するソフトウェアであり、一種のインタプリタといえる。通常のインタプリタが解釈するのが高水準言語であるのに対して、仮想機械の場合は機械語そのものである。

最初に提案された（ハードウェアを模した）仮想機械としては、IBM VM/370 があげられる。近年では、パソコン上で動作する仮想機械でパソコン自身を模するものとして、VMWare、Microsoft VirtualPC、Xen などが盛んに使われ、これらは、複数のオペレーティングシステムを同一のコンピュータ上で同時に使用することを可能とし、利用者の利便性を向上させている。

これらの仮想機械は、オペレーティングシステムと同等にハードウェア上で直接動作するものもあれば、オペレーティングシステム上のアプリケーションソフトウェアとして動作するものもある。また、仮想機械が動作する CPU と仮想機械が仮想化の対象とする CPU が同一の場合もあれば異なる場合もある。

実際に存在する、もしくは、過去に存在したハードウェアを仮想化する仮想機械の他に、実際には存在しない仮想的なコンピュータを仮想化した仮想機械も存在する。これは、高水準言語を対象とするインタプリタに近い存在であり、両者



図 13 パソコン上の仮想機械ソフトウェア

の違いは、対象が機械語であるか高水準言語であるかという点である。このような仮想的なコンピュータを実現した仮想機械として現在代表的なものとしては、JVM (Java Virtual Machine) があげられる。プログラミング言語 Java の標準的な言語処理系 (すなわち言語提唱者であった Sun Microsystems が提供するもの) は、Java で書かれたプログラムをいったん Java バイトコードという仮想的な機械語にコンパイルし、それを JVM の上で実行する。JVM を実装するだけで Java バイトコードのプログラムを動作させることができ、Java プログラムから Java バイトコードプログラムへのコンパイラは共通のものが利用できるというのが、この方法の最大の利点である。現在では、パソコンをはじめとして、携帯電話、そして、スマートカードに至るまで多種多様なコンピュータで JVM が実現されている。

この種の、仮想機械が実行機械とする仮想的なコンピュータの機械語は、バイトコードと呼ばれる。Java のようにバイトコードにコンパイルして、仮想機械で実行するという方法は、Java 以前より様々なプログラミング言語で使われてきた。

仮想化はソフトウェアに留まらず、最近では CPU のようなハードウェアにおいても仮想機械というアイデアが使われている。たとえば、インテル社の Pentium Pro から Core2Duo に至る CPU では、それ以前の CPU の機械語と同じものを使用しているが、実際には μ OPs と呼ばれるハードウェアで処理しやすい機械語に変換してから実行している。したがって、CPU 自体が既存の CPU を模倣する一種の仮想機械となっている。本来であれば、CPU のハードウェア的な特徴を反映した機械語を定義して、性能を十分引き出せるようにコンパイラを作成した方が、プログラムの高速化が期待できると考えられる。

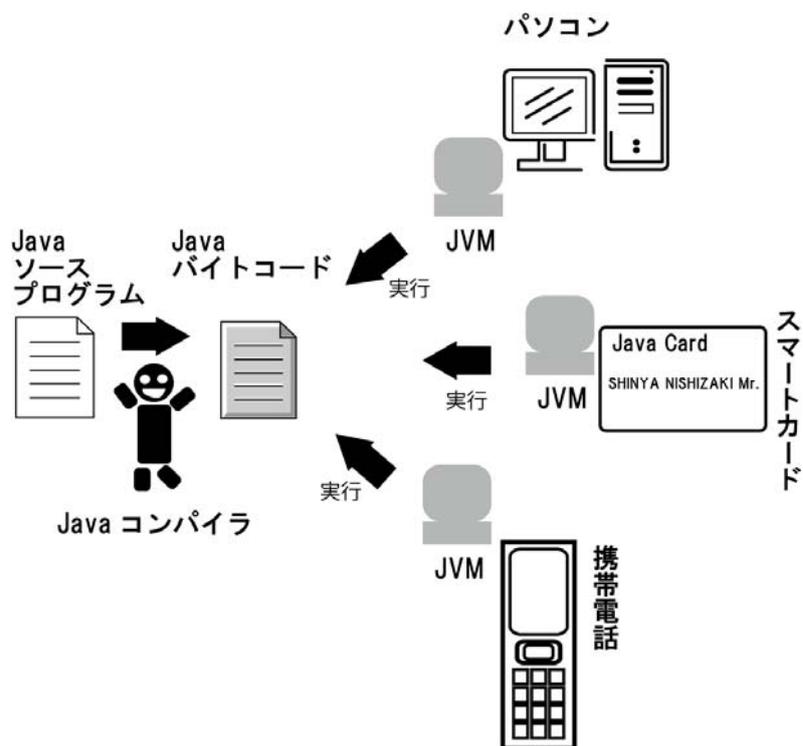


図 14 さまざまなコンピュータ上の JVM

しかし、既存のソフトウェア資産を継承する方が重要であるために、このような手法が取られてきたが、実装技術の進歩により、十分な成果が得られた。この二つのアプローチのいずれが有利なのかは、まだまだ議論の余地がある。

3.5 コンピュータネットワーク

コンピュータの発展（高速化・大容量化）に加え、それらの間で情報を享受する場合の基盤となるコンピュータネットワーク（以下、ネットワーク）も進歩・拡大している。ネットワークにより、手許に情報機器があれば、誰もが瞬時に情報を入手すること、個人で社会に向けて情報を発信すること、仮想的な空間の中で共同作業をすることなどが可能になっている。ここでは、そのようなネットワークの仕組みや特質、それを実現する技術などについて述べる。

3.5.1 情報の享受

情報を享受する上では、送信者と受信者の双方において、情報の享受に関わるあらゆるものについて共通の理解が必要である。たとえば、人間同士のコミュニケーションでは、話し手と聞き手の双方が理解できる言語（たとえば、日本語）が使用され、お互いにその意味を知っているであろう単語により、会話の内容が構成される。人間同士であれば、未知の単語が用いられた場合には、意味を確認

するためのやりとりも行われる。人間同士の場合には、特に事前の取り決めがなくとも柔軟に対処し、コミュニケーションを進めながら、お互いに共通の理解を構築していくことも可能である。

コンピュータ間の通信による情報の享受においても、人間同士のコミュニケーションと同様に共通の理解が必要である。人間と異なるのは、その共通の理解を事前に明示しておかねばならないことである。どのようなサービスについてのやり取りであるのか、そこでは、情報がどのようなデジタルデータとして表現されるのか、やりとりの開始、終了、やり直しはどのような手順で実行するのか、相手をどのようにして特定し、そこまでの通信経路（含：伝達方法）を決めるのか、データはどのような信号として送られるのか、このようなことを事前に定めた規約にしたがって、通信が実現されている。

3.5.2 通信機能の階層化

封書による情報伝達を例に、通信回路の階層化を考えよう。送り手は、便箋に内容を記述して封筒に入れ、封筒には宛先を記入し、ポストに投函する。投函された封書は、郵便局などの事業者により受取人の郵便受けに配達される。封書の作成が終了すれば、ポストへ投函する作業は、封書の作成者本人である必要はなく、他の人に依頼することができる。封書をポストに投函するよう頼まれた人は、迅速に封書を投函すれば、どのポストに投函してもよく、受取人の郵便受けまで配達する必要もない。配達を依頼された事業者は、迅速に正確に配達すれば、どのような手段や経路で配達しても構わない。

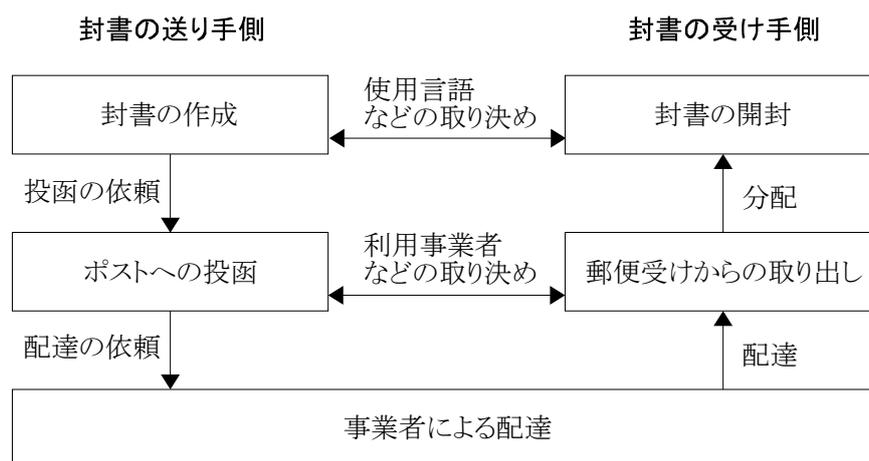


図 15 手紙による情報伝達の階層化

このように、封書による情報伝達は、(1) 封書の作成、(2) ポストへの投函、

(3) 配達、のような独立した役割に分割された階層構造とみなすことができる。これと同様に、ネットワークにおける通信機能も、ブラックボックスとして扱えるような独立した役割ごとに分割し、階層構造を形成する。はがき、手紙、小包のように、相手に送るものが変わっても、配達経路は共通に利用できるように、ネットワークの通信機能についても、下位に位置する階層は、アプリケーションに共通して利用できる機能が配置される。上位に位置する階層に配置される機能は、アプリケーションに固有の機能になる。

封書の送り手は、受取人が封書の内容を読むことを期待している。そのために、互いに理解できる言語を用いて内容を記述する。ネットワークにおける通信においても同様に、役割ごとに分割された各階層は、送受信双方にそれぞれ対応する役割の階層が存在し、それらの間でその階層としての役割を果たすための約束事が仕様として定められている。この約束事を通信規約（通信プロトコル）という。

3.5.3 階層化による利点

通信機能を階層化する利点の一つは、自層より下の階層における動作の詳細を知る必要がなくなることである。例えば、ポストに投函された封書が、受取人に届けられることさえ保証されれば、どのような手段と経路で配達されるかを知る必要がない。これと同様に、電子メールなどもソフトウェアからメール送信の指示を与えるだけで、配送の詳細を知ることなく利用できるようになる。

もうひとつの利点は、各階層が独立し、その役割や処理内容が定められているため、その階層の仕様を満たせば、（原理的には）他の階層へ影響を与えることなく、新しい実装や改良などが可能であることである。例えば、荷物を送る場合には、小包を作成する作業や、事業者の窓口への持ち込みなど封書の場合と異なる処理が必要である。これらに対応する階層で新しい処理方法として定義し、封書の場合と置き換えることで、配達機能としては封書の場合と同じものが利用できる。これと同様に、ネットワークを利用した新しいサービスが生まれた場合でも、そのサービス固有の機能を有する階層のみ実装することで、他の機能をそのまま利用できるようになる。

3.5.4 ネットワークの形成

共通の通信プロトコルを使用し、機器間の通信路を設けることにより、原理的にはその機器間での通信が可能になる。機器数が増加すると、任意の機器間を1対1で接続する通信路を設けることは次第に困難になるため、幹線となるような通信路か、または、それに相当する装置に機器を接続し、すべての機器で通信路を共有したネットワークを形成する。

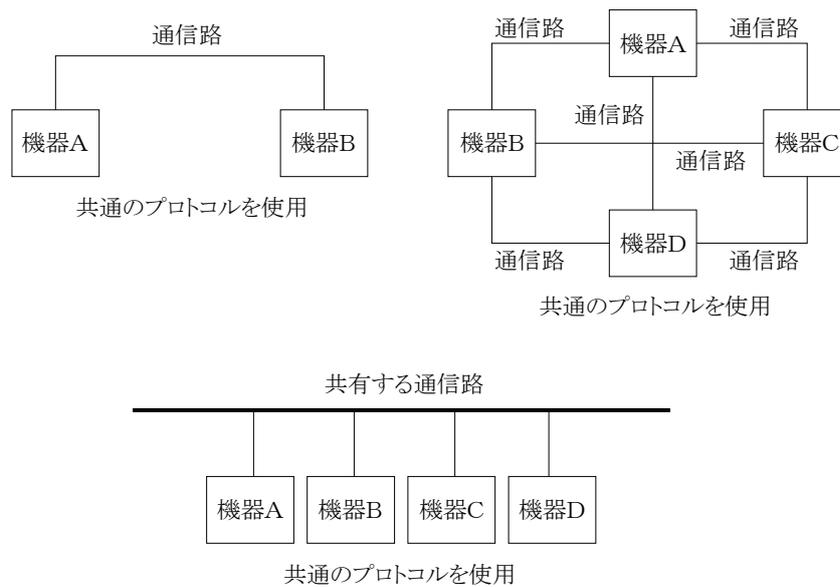


図 16 ネットワークの形成

共有される通信路には、そこでの通信に用いられるすべての信号が流通する。そのため、信号を送出する権利を制御して、通信路における信号の伝送を制御する仕組みや、機器ごとに定められた識別番号を使用して、通信相手を特定するための仕組みなどを定めた通信プロトコルが用いられる。このような仕組みのもとで、室内、フロア、建物全体などの比較的狭い範囲において、機器が接続されるネットワークをローカルエリアネットワークという。

3.5.5 インターネット

ローカルエリアネットワークを相互に接続し、ネットワークのためのネットワークとして、世界中に拡大したものがインターネットである。インターネットで事実上の標準規格として用いられている通信プロトコルでは、通信相手を特定するための識別番号として、IPアドレスを使用する。

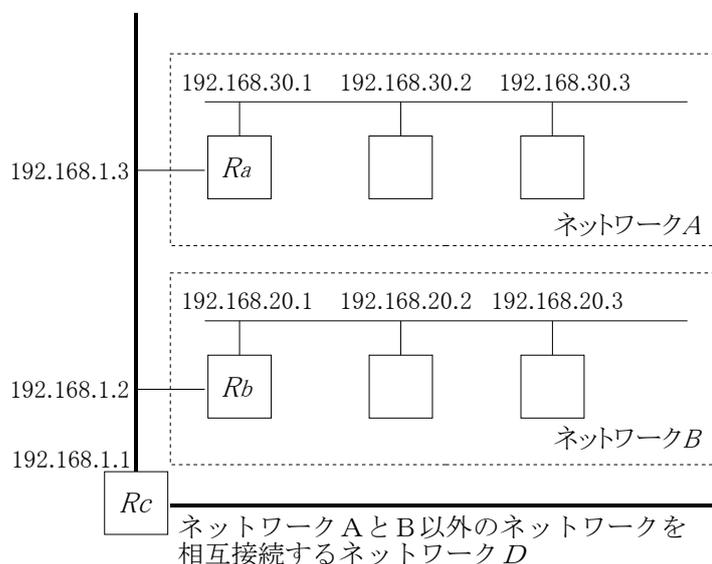
IPアドレスは、8ビットの数値4つからなる32ビットの数値である。10進数で表記すると、0から255までのいずれかの数値（8ビットで表現される数値）4つを「.（ドット）」で接続し、192.168.1.10のようになる。

インターネットでは、データを少量のパケットと呼ばれる単位に分割し、通信相手のIPアドレスなどの情報を付加して伝送する。複数のネットワークの接続点にある機器では、パケットに付加されたIPアドレスに応じて、そのパケットをどのネットワークに転送するのかを決定する。このような役割をする機器がインターネット全体で協調して動作し、間接的に接続された機器間の通信が実現されている。

ホームページを閲覧する場合、通常は 192.168.1.10 のように、IP アドレスのような数値を利用せず、www.science-for-all.jp のような文字列でホームページを提供しているコンピュータを特定している。このようにインターネットで機器を特定するためにつけられる名前をドメイン名という。通信の制御は、前述のように IP アドレスで行われるが、人間にとってはドメイン名の方が扱いやすい。そこで、ネットワーク上にドメイン名と IP アドレスを相互に変換する仕組みを設け、インターネットを利用するアプリケーションではドメイン名を使用できるようになっている。

例 10 : 間接的に接続された機器間の通信

ネットワーク A と B を
相互接続するネットワーク C



上の図は、機器 R_a がネットワーク A と C を接続、機器 R_b がネットワーク B と C を接続、機器 R_c がネットワーク C と D を接続している様子を示している。このとき、パケットの宛先 IP アドレス（以下、宛先）について次のように設定すると、間接的に接続された機器との通信が実現される。なお記号 * は 1 から 3 のいずれかを表すものとする。

- R_a はパケットの宛先が 192.168.30.* であれば A 側に転送し、それ以外は R_c に転送する
- R_b はパケットの宛先が 192.168.20.* であれば B 側に転送し、それ以外は R_c に転送する
- R_c はパケットの宛先が 192.168.1.* であれば C 側に転送し、それ以外は D 側に転送する

このような設定を施してネットワークの相互接続を広げることにより、インターネットにおける通信が実現される。

3.6 情報セキュリティ

情報セキュリティの目的は、安全に情報を活用することにある。安全性を重視することにより、活用の利便性を損ねることや、利便性を重視することにより、安全性がおろそかになるようなことがないように、両者のバランスをとることが重要になる。ここでは、情報セキュリティに関する基本的な考え方とその技術について述べる。

3.6.1 情報セキュリティの重要性

情報セキュリティとは、基本的には、許可されたものだけに情報の利用を認める機密性、情報やその処理内容が正確であり完全である状態を維持する完全性、許可されたものが情報を利用するときには、いつでも利用できるようにする可用性を維持、確保することである。

たとえば、ショッピングサイトなどでは、通常、利用者の個人情報に加え、クレジットカードの番号や過去の購入履歴などを蓄積している。これらの情報が、万が一、外部からも自由に閲覧できる状態であれば、利用者のプライバシーは守られず、このサイトを経営する企業は信頼を失うであろう。また、注文した商品の種類や個数が正しく記録されなかったり、支払い金額を正確に計算できなかったりする場合には、利用者は安心して商品を購入することができなくなる。一方で、情報の外部への漏洩を過度に恐れるあまり、利用者に対してその人自身の購入履歴の閲覧や、注文のキャンセル、追加などの機能まで制限すると、利用者はネットショッピングの便利さを享受できなくなる。

このように、情報セキュリティは、企業や組織の機密情報を守るという意味合いだけでなく、個人の生活の利便性にまで影響するものである。

3.6.2 情報セキュリティの対象

情報セキュリティの考え方の中では、情報を利用する主体となるものは、人間だけに限定されない。自動的に情報を読み取って処理するコンピュータや、そのような働きをするプログラムなども利用主体である。

同様に、保護対象となる情報も、コンピュータ上に保存されているデータだけに限定されるものではない。CD、DVDなどのメディアや、コンピュータ、ネットワークなどもその対象となる。より一般には、印刷体の資料などデータ化されていない情報も対象となる。このような情報全般をさして、情報資産と表現することもある。

3.6.3 認証

情報資産の利用を制御するためには、利用主体を特定する必要がある。この手続きが認証である。個人を認証する技術として一般的なものは、暗証番号やパスワードなど、本人しか知りえない知識を用いる方法である。システムにあらかじめ記録されている情報と、キーボードなどから入力される情報を照合して認証が行われる。暗証番号やパスワードの人力操作を盗み見される危険性を回避するために、認証のための情報を記録した ICカードなど、本人しか持ち得ないものを利用する方法もある。所有物は紛失・盗難などの危険があるために、本人の身体の一部に由来し、偽造することが難しく、その人固有の特徴を利用して認証を行う方法もある。これを生体認証という。生体認証には、指紋、指または手のひらの静脈パターン、眼球の虹彩パターンなどが使われる

3.6.4 データの暗号化

データを取得しても、その内容を容易に読み取られないようにする技術に暗号技術がある。あるデータに対して、定められた方法とその方法を特徴付ける別のデータを用いて変換を施し、その逆変換の方法を知らないものには、もとのデータを容易には取り出せないようにすることを暗号化という。

変換のための方法を暗号アルゴリズムとよび、変換の方法を特徴付ける別のデータのことを鍵という。暗号化する前のデータのことを平文、平文を暗号化したものを暗号文、暗号文からもとの平文を取り出すことを復号という。暗号アルゴリズムと鍵がわかれば、復号が可能である。

簡単な暗号の例を一つ紹介しよう。暗号アルゴリズムとして、英字をアルファベット順で x 文字後ろの文字にずらす、というものである。たとえば、 $x = 1$ のとき、 a は b に、 b は c に置き換えられて暗号化する。 z は最初に戻って a になるのである。information という平文は、jogpsnbujpo という暗号文になる。 $x = 1$ であることを知っていれば、暗号文の英字から平文を復号することができる。

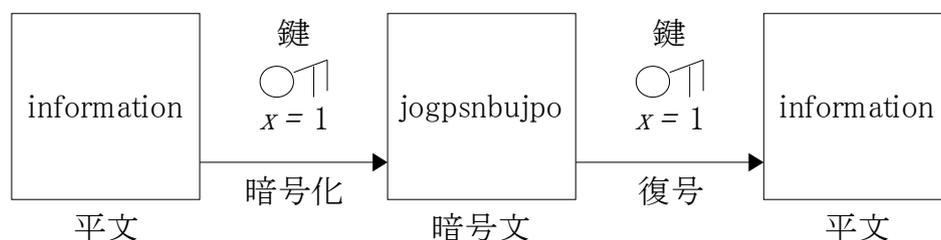


図 17 暗号化の一例

暗号を用いると、対象となるデータを暗号化し、利用を許可するものだけに復号に必要な鍵を付与すれば、その鍵を与えられたものだけがデータを利用できるという意味で、データの利用を制御できる。また、データを暗号化するものを送信者、復号に必要な鍵を付与されたものを受信者として暗号文を伝送することにより、両者の間で暗号を用いた通信が可能になる。

3.6.5 共通鍵暗号方式

上の例のように、暗号化の際と復号の際に同じ鍵 (共通の鍵)を用いる暗号方式は、共通鍵暗号方式と呼ばれる。使用する鍵の所有者に関する知識 (誰がその鍵を所有しているか知っているということ)に基づいて、互いに相手を特定することができる。たとえば、この方式でメッセージを暗号化し通信する場合、復号に使用した鍵から、誰がメッセージを暗号化したのかを特定できると同時に、確かに自分宛へのメッセージであることが理解できる。

3.6.6 公開鍵暗号方式

暗号化と復号の際に、1対のものとして生成された公開鍵と秘密鍵の2つの鍵を使い分ける公開鍵暗号方式がある。秘密鍵は、鍵を生成した本人が所有し、漏洩することのないように管理する。それに対応する公開鍵は、誰もが使用できるように公開する。この暗号方式には、次の二つの特徴がある。

- ▶ 公開鍵で暗号化された暗号文は、それと対をなす秘密鍵だけが復号可能である
- ▶ 秘密鍵で暗号化された暗号文は、それと対をなす公開鍵だけが復号可能である

この方式で、メッセージを暗号化し、通信する場合を考えよう。送信者の立場では、受信者の秘密鍵は使用できないことと、送信者の公開鍵で暗号化すると、送信者以外の者は復号できないことから、暗号化に使用できるのは、受信者の公開鍵と送信者の秘密鍵になる。

送信者が受信者の公開鍵によってメッセージを暗号化すると、復号可能な者は、受信者の秘密鍵を所有する者、すなわち、受信者本人に限られるため、特定の受信者以外に内容を知られることなく通信が可能となる。しかし、メッセージの受信者は、そのメッセージが誰でも使用できる公開鍵により暗号化されているため、送信者を特定することができない。

一方、送信者が自らの秘密鍵でメッセージを暗号化すると、誰でも使用できる送信者の公開鍵で復号可能であるから、内容を保護することも、受信者を特定することもできない。しかし、その暗号文は送信者しか所有しえない秘密鍵により

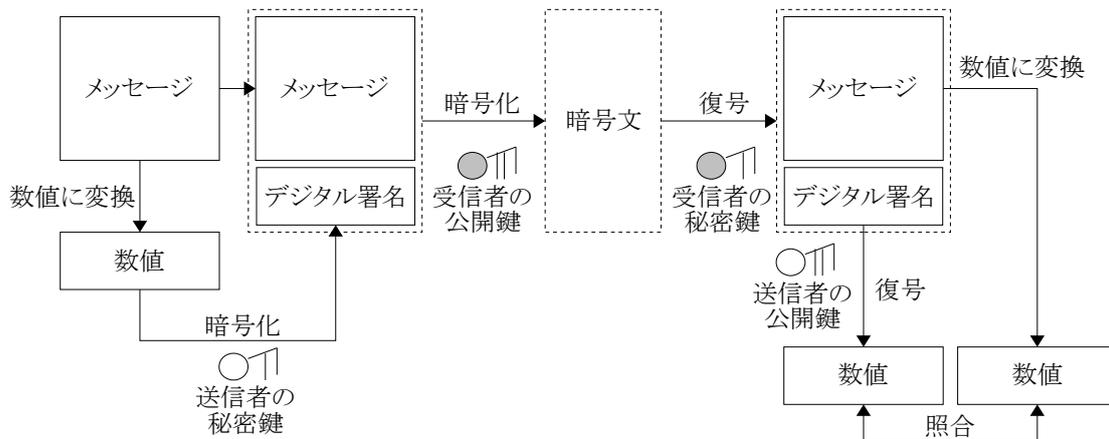


図 18 公開鍵暗号方式

暗号化されたものであることから、受信者はメッセージの送信者を特定することが可能となる。

この特性を利用したものがデジタル署名である。デジタル署名としては、一般には、メッセージを数値に変換し、その値を送信者の秘密鍵により暗号化したものが用いられる。送信者はデジタル署名をメッセージに添付し、受信者の公開鍵で暗号化することにより、受信者に対して自分が送信者であることを証明できると同時に、受信者を特定してメッセージを送付することができる。

3.6.7 ハイブリッド暗号方式

共通鍵暗号方式と公開鍵暗号方式を組み合わせ、両者の利点を活かした暗号方式である。

共通鍵暗号方式では、事前に鍵の交換が必要である。鍵の漏洩を防止するという点からは、(鍵は電子的なデータであるとはいえ)ネットワークを介して交換することは安全ではない。また、知らない相手や遠隔地の相手との鍵の交換は、一般には容易ではない。

一方、公開鍵暗号方式では、公開鍵は誰でも利用できるものであるから、漏洩などを心配する必要はない。また、公開鍵が自分のものであるということを第三者機関に証明してもらえらる限り、相手と直接的に鍵を交換する必要がない。

さらに、共通鍵暗号方式による暗号化・復号の処理時間は、公開鍵暗号方式によるそれと比べて短時間である。これらを考慮し、平文の暗号化には共通鍵暗号方式を用いて、そこで使用する鍵の交換に公開鍵暗号方式を用いるのが、ハイブリッド暗号方式である。

3.6.8 暗号による認証

共通鍵暗号方式では、事前に鍵を交換した相手との間で、暗号文をやり取りすることにより、互いに認証が行われる。鍵の交換を行う過程において、本人確認などの社会的な手続きを終えていることが前提となり、その上で暗号による認証を真正なものとして取り扱える。

公開鍵暗号方式では、相手に自分の公開鍵を使用させることで、自分を特定させ、自分は相手の公開鍵を使用して相手を特定する。ともに、公開鍵に対応する秘密鍵の所有者が、間違いなくその公開鍵の所有者であることを前提としている。技術的には、秘密鍵と公開鍵の生成、ならびに公開鍵の公開は、社会的な手続きを要しないため、公開鍵の真正性を社会的に担保する必要がある。

これを実現する仕組みがPKI (Public Key Infrastructure) である。これは、秘密鍵に対応する公開鍵の所有者であることを社会的な手続きに基づいて証明してもらい、その証明機関を信頼することで、公開鍵の真正性を保証し、その電子的な証明書をネットワーク上で発行するものである。

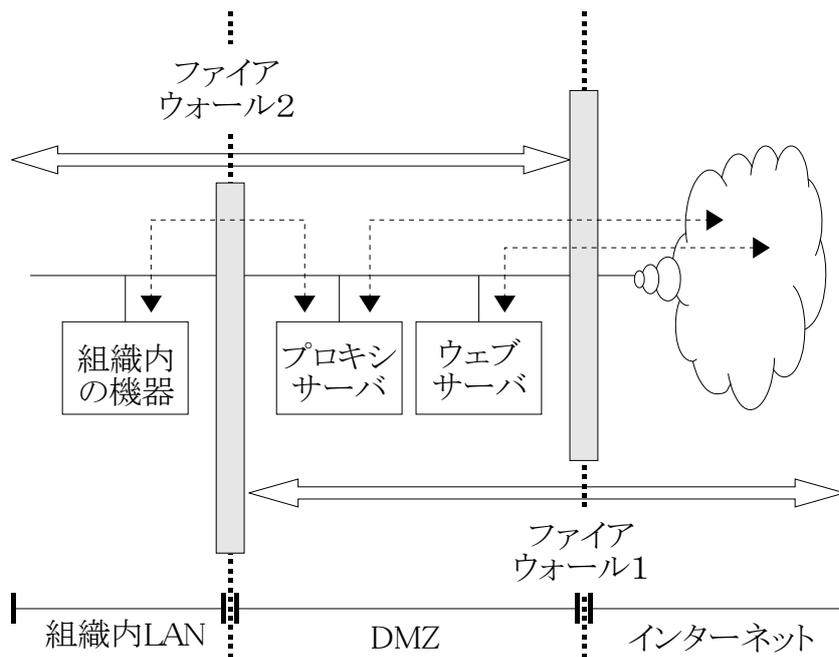
3.6.9 ネットワークに対する情報セキュリティ対策

ネットワークに接続することによって、様々なサービスを受けたり、提供したりすることができるようになる一方で、そのサービスを妨害したり、情報を盗み出したりする攻撃を受ける恐れがある。また、コンピュータウイルスなどに感染することにより、情報が意図せず漏洩することなども起こりうる。このような危険から情報資産を保護するために、インターネットとの接続点を通過する通信を制御する仕組みがファイアウォールである。

ファイアウォールは、許可した種類の通信のみを中継したり、特定の機器のみ通信を許可したりすることにより、攻撃や情報漏洩を防ごうとするものである。図 19 は、一般利用者向けにホームページを公開している企業などが、組織内部のローカルエリアネットワークとインターネットとの接続点にファイアウォールを設置する場合の概念図である。

このような構成にすることにより、インターネットからは DMZ に接続された機器までは攻撃の危険にさらされるが、ファイアウォール 2 によって通信が遮断されるため、組織内 LAN に影響が及ばないようにすることが可能である。また、組織内 LAN からは、ファイアウォール 1 によって通信が遮断されるため、インターネットへの意図せぬ情報漏洩を防止できる可能性がある。

一方、組織内の利用者は、このような仕組みにより、インターネットへの直接のアクセスはできなくなる。このため、DMZ に接続されたプロキシサーバと呼ばれる機器を経由して、インターネットにアクセスする方法が提供される。プロキシサーバにおいては、通信の種類や相手先などに基づく制限に加え、データの中



この例では二つのファイアウォールが用いられている（実際には、一台の機器で二役させることも可能）。

ファイアウォール1は、インターネットとそこからアクセスされる機器を接続するDMZ（緩衝地帯、あるいは、非武装地帯）と呼ばれるネットワークとの接続点に設置する。ファイアウォール2は、DMZとインターネットからのアクセスを禁止する組織内ローカルエリアネットワーク（組織内LAN）との接続点に設置する。ファイアウォール1では、インターネットとDMZの間の通信のみ許可し、組織内LANからインターネットへ向かう通信を遮断する。ファイアウォール2では、DMZと組織内LANの間の通信のみ許可し、インターネットから組織内LANへ向かう通信を遮断するのである。

図 19 ファイアウォールの設置例

身を確認して、通信を細かく制御することが可能である。具体的には、送られてきたデータの中に、コンピュータウイルスが混在していないかどうかや、その組織において閲覧するには不適切なデータが含まれていないかなどを確認することができる。

このように、ファイアウォールを使用することで不要な通信を遮断し、インターネットからの攻撃に対する組織内部のネットワークの安全性を高めることができるようになるが、完全に防御できるものではないことに注意が必要である。図の構成でも、Webサーバがインターネットからのアクセスを受け付ける以上（ファイアウォール1がその通信を許可する以上）、それを悪用した攻撃に対応する

ことは難しい。同様に、プロキシサーバを経由して、組織内 LAN からインターネットにアクセスできる以上、プロキシサーバが中継する通信方法によって、組織内部からの情報漏洩の危険は残るものと考えerほうがよい。

3.6.10 個人・組織・社会における情報セキュリティ対策

個人認証の技術や二値符号技術など、技術的な情報セキュリティ対策が進歩しても、それを正しく理解し活用できなければ、安全な情報の利用は難しい。また、そのような技術的な対策だけでは対応できない部分がある。

たとえば、パスワードによる認証を設けている場合、それによって情報を保護できるのは、パスワードが漏洩しないように管理していることが大前提である。パスワードを厳重に管理し、本人以外に知るものがないことが確実な場合でも、その人だけが利用できる情報をコンピュータのモニタに表示したまま席をはずしてしまうと、その情報は漏洩の危険にさらされていることになる。着席している時でさえも、オフィス内部の机の配置などのために、背後を人が通るような状況では、覗き見されるなどの危険は避けられない。

また、企業などの会計情報を処理するシステムでは、関連する法令の改正などに伴って、処理内容を変更する必要がある。このときには、法令の解釈や会計処理に詳しい人との共同作業によって、法令に準拠した正しい処理を行うシステムへと改訂し、完全性を維持するよう努めなければならない。

さらに、災害や機器の故障などにより、情報の利用を制限される恐れがある。場合によってはデータが失われることもあるため、バックアップシステムを遠隔地に設置したり、システムやネットワークを冗長化したりする必要がある。予期せぬ停電なども想定した対応も情報セキュリティ対策のひとつとなる。

3.7 基盤となる応用技術

ここまではコンピュータ・システムの基礎となる仕組みを述べてきたが、この節では、その他の情報科学技術の基盤となる応用技術について紹介する。情報科学技術ロールケーキ（巻頭の図）のスポンジ部分に記載されている各要素技術の簡単な解説である。

3.7.1 データベース

情報化の進展と記録技術の進歩に伴い、社会の各所で大量のデータが蓄積されるようになってきている。これらを有効に効率よく活用するためには、組織的にデータを管理する必要がある。ある目的のもとで収集したデータを容易に活用できるように定型化・組織化し、データの追加、更新、削除、検索などデータの操作に

関わる処理を行う仕組みがデータベースである。データ操作を効率的に行うために、データの論理的な格納方法を定めたものがデータモデルである。

代表的なデータモデルのひとつに、関係モデルがある。直感的には、表形式でデータを整理するようなモデルである。表の各行が、1件のひとまとまりのデータに対応する。表の各列は、ひとまとまりのデータに含まれる項目に対応し、属性とよばれる。関係モデルに基づいて作られるデータベースが関係データベースである。関係データベースでは、格納している表から特定の行(たとえば、ある特定の郵便番号の人だけ)や特定の列(たとえば、氏名と電話番号だけ)を抽出して新しい表を作成したり、他の表と組み合わせ、両方に格納されている行(たとえば、他のグループの住所録とつき合わせて、両方に属す人だけ)や、いずれか一方に格納されている行からなる新しい表(たとえば、ふたつのグループの住所録を統合したものになる)を作成したりすることにより、必要なデータを抽出できるようになっている。

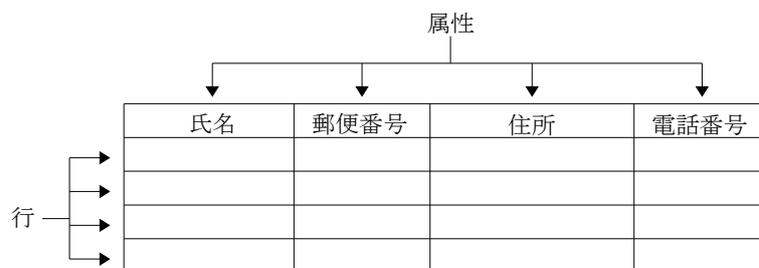


図 20 住所録を例にした関係データベースの概念図

3.7.2 データマイニング

情報技術の進歩により、インターネットなどからのデータ収集が容易になり、収集したデータの保存に要するコストも低下したことによって、大量のデータを取り扱える環境が整ってきた。また、処理速度の高速化やアルゴリズムの改良などにより、大量のデータを事前に選別することなく分析できるようになった。これらを背景として、無意味に思えるデータの組み合わせなどに対しても分析を行い、未知の関係や法則を導き出そうとする手法がデータマイニングである。

理論的な基盤は統計的な分析手法であるが、対象固有の現象や特徴を反映した処理を加えるなどの工夫により、処理結果の選別を行う場合がある。分析結果には、既知の法則や関係も多く見出される。また、新しい法則や関係が発見されたとしても、それが本当に意味のあるものかどうかや、役に立つものであるかどうかは、別に検証されなければならない。

3.7.3 文書処理

インターネット上には、人間の処理能力を超える大量の文書が蓄積されている。それらの中から、利用者が必要とする情報を掲載した文書を効率よく検索するときには、適切な検索語を指定し、対象とする文書のなかから、その検索語を含む文書を抽出する処理が行われる。対象となる文書の内容を示すようなキーワードを見出しとしてあらかじめ設定しておき、その箇所のみを対象として検索語との照合を行う方法や、文書に含まれているすべての語を対象として検索語との照合を行う方法などがある。

既存文書を処理するこのような技術に対し、新たに作られる文書については、検索や要約の対象となることを前提として、それらの処理が効率よく効果的に行われるようにするために、文書を作成することも行われている。そのように作成された文書では、語句、文、段落などに、その意味・内容や、文書中での役割などを示すためにあらかじめ定められたタグが付与される。タグが付された部分をどのように処理するかを定めたプログラムを用いることにより、文書の閲覧、印刷物の生成、管理などが行える。

3.7.4 検索技術

文書データを対象とする場合、指定された検索語とデータに現れる語を逐次照合し、一致する部分を探し出すことが最も簡単な検索手法である。データ全体に現れる語の索引を別に生成して付与したり、データの内容を示すキーワードなどをメタデータとして付与し、それを手がかりとして目的とする文書を検索することが行われる。文書データ以外にも、画像データや音声データに対して、その内容に関するメタデータを付与することにより、キーワードを用いた検索が可能である。

二種類のデータが同一のものであるかどうかを判定するために、それぞれのデータを数値化し、その数値による比較を行うことがある。このアイデアを応用し、メタデータを使用せずに、画像データや音声データの検索を行うこともできつつある。キーワードのかわりに、検索したい画像データや音声データと特徴が類似しているデータを指定し、それらを数値化して得られる値と、検索対象となるデータの値とを比較することによって、目的とするデータを探し出そうとするものである。二つのデータの類似性をうまく反映する数値化の手法を構成することが必要である。

3.7.5 信号処理

外界の状況を観測する装置として、様々なセンサがある。センサからの信号を変換、加工する技術を信号処理という。アナログ信号をデジタル信号に変換する処理は、信号処理のひとつのアプリケーションである。これ以外にも、ノイズの除去、特徴の抽出、数学的な変換など、様々な応用がある。

信号処理では、処理対象となる信号を入力し、目的とする変換が行われる電子回路を用いることがある。コンピュータで行う信号処理は、入力信号をデジタルデータに変換し、プログラムによって変換が行われる。出力信号は、変換後のデジタルデータから再構成される。

コンピュータで行われる画像処理では、入力画像として撮像系からの信号をデジタルデータに変換してコンピュータに入力し、(そのままデジタルデータとして記録することも含め)数学的な変換を施したのち、再び画像に変換して(すなわち、デジタルアナログ変換が行われ)その結果を表示する。音声処理も同様に、音質、速度、音程の変更など、デジタル化された音声データに対して変換を施し、アナログ変換を行ったのち、再び音声として出力する。

3.7.6 可視化

信号処理においては、対象となる信号を人間の視覚や聴覚によって知覚されるものに限定していない。つまり、人間が直接、見たり聞いたりできない信号も対象となる。それらを人間が知覚できる信号に変換し、表示する技術が可視化である。処理対象となるもとの信号がアナログ信号、デジタル信号のいずれであっても、入力信号や処理結果となる信号をデジタルデータに変換することができれば、人間が知覚しうる色や書に対応付けることが可能である。これがコンピュータによる可視化である。そこでは、データを画像情報として表現するコンピュータグラフィックスや、データを音声情報として表現する音声合成などの技術を利用することができる。信号処理と関係のない対象であっても、それをデジタルデータで記述することが可能であれば、表現形式を適切に定めることにより可視化することは可能である。

3.7.7 シミュレーション

自然現象や社会現象などの構成要素をモデル化し、ある状況設定のもとで生じる現象の予測・分析などを行うことをシミュレーションという。特に、コンピュータ上でモデル化し、数値解析的な手法により現象を予測・分析することをコンピュータシミュレーションという。解析的には解けない現象、実験することがで

きない現象、状況を変更して繰り返し実験することが困難な現象などは、コンピュータシミュレーションが有効である。実行状況や結果を可視化技術により提示することも可能である。

シミュレーションに人間から割り込みを認め、状況設定の変更を動的に受け付ける仕組みをシミュレーターという。フライトシミュレータなど、実際の機器操作により状況設定の変更を受け付けるような場合には、その操作によりもたらされる結果を機器の振る舞いへとフィードバックし、現実感のあるシミュレーションが行える。実機の操作と変わらない操作感覚を実現するために、現実感のあるグラフィックスを生成する技術や、実機との時間遅れなどが生じないように高速に演算、制御する技術が必要である。

3.7.8 パターン認識

信号から信号へ変換・加工する技術が信号処理であることに対し、入力信号のクラスタリングや、記号記述への変換を行う処理は、特にパターン認識とよばれる。たとえば、画像中で意味のある特定の領域を抽出する処理や、画像中の被写体は何であるかを特定するような処理などは、画像認識という。画像中から人物を同定したり、文字が書かれた印刷物などの画像から、文字情報を抽出したりする処理などがある。また、音声を対象にして話者を特定したり、発話内容を文字列に変換したりする処理は音声認識という。これにより、声によって機器を制御することなどが可能である。

3.7.9 人工知能

人間が行っている知的な活動や知能の働きをコンピュータ上で実現しようとする試みが人工知能である。顔画像による認証などは、人間の持つ視覚情報処理能力の一部を画像認識技術として実現したものである。機械翻訳などは、人間の持つ言語活用能力の一部を模倣しようとしたものである。これらの他には、知識に基づいて判断・推論し、思考を支援するための情報を提供するシステムや、統計的な分析手法に基づいて、知識や判断規則などを抽出する手法がある。

3.7.10 マルチメディア

文字、音声、画像 (動画像を含む)などは、情報を表現するメディアであり、それぞれ表現できる情報の種類が異なっている。いずれのメディアで表現される情報もデジタル化されることにより、プログラムという統一的な方法で取り扱うことが可能である。このように、様々な種類の情報をデジタルデータで記述し、プログラムによって処理する仕組みをマルチメディアという。

マルチメディアデータの処理には、通信プロトコルに類似した階層構造がある。下位層では、メディアごとの処理は不要であり、データ全体を誤りなく処理する機能が必要である。中間層では、各メディアの内容を取り出すために、データのどの部分がどのメディアに対応しているかを区別する処理が必要である。そのために、そのような区別をするためのデータが埋め込まれており、プログラムによって読み取られるようになっている。上位層では、メディアを特定されたデータに対し、それぞれ固有の処理が行われ、内容が取り出される。たとえば、マルチメディアデータのうち、文字情報として区分された部分は、文字コードとして解釈され、文章が取り出されることになる。

3.7.11 HCI (Human Computer Interaction)

コンピュータの使いやすさを提供する仕組みのひとつがユーザインタフェースである。なかでもグラフィカルユーザインタフェース (GUI)は、メニュー、アイコン、ボタンなどの部品を用いて、利用できる機能や現在の状態などを視覚情報として提示するものである。キーボードやマウスなどを通して、それらの部品を操作することが、実際のコンピュータ操作につながっている。

ネットワーク環境においては、ネットワークを通じて様々なサービスが提供される。サービスへのアクセスには、GUIを持つウェブブラウザが用いられる。サービスの窓口となるのは、ウェブページであることが多く、その場合には、ウェブページがユーザインタフェースの役割を果たす。

キーボードやマウス以外からコンピュータを操作するユーザインタフェースもある。たとえば、音声認識技術を応用することにより、音声でコンピュータを制御することや、音声合成技術により、文書の内容を読み上げることが可能である。また、画像処理、画像認識技術を用いることにより、コンピュータの画面に向けられた利用者の視線に基づいて、次に行われる操作を予測することも可能である。このような機器と人間との間のやり取りを様々な角度から分析し、ユーザインタフェースの設計や実装に活かす取り組みが HCI である。

3.7.12 組み込み技術

情報機器や家電製品にとどまらず、電子制御を行う製品の多くには、部品のひとつとして CPU が搭載されている。この CPU で、その製品に必要な特定の機能を実現するプログラムの開発技術が組み込み技術である。

CPU はプログラムによってその動作を自由に定められることから、機器本体の回路などを変更することなく、プログラムの変更・修正により機能を追加したり、不具合を修正したりすることが可能となる。基本的には、特定の機能を実現する

ものであるから、多機能な製品は、複数の組み込みシステムの集合体となることがある。

組み込みシステムの CPU やメモリには、製造コストや製品の大きさなどの面から、一般のコンピュータに搭載されているような CPU やメモリと同程度の性能・容量は望めない。そのため、プログラムの開発にあたっては、動作速度や使用可能なメモリ容量などの制約が多くなる。そのような制約があるなかで、高機能の組み込みシステムを開発するには、効率のよいアルゴリズムの設計など理論的なアプローチが必要になる。

3.7.13 ユビキタスコンピューティング

CPU の小型・低価格化と、組み込み技術の進展により、人間の生活空間のあらゆる場所にコンピュータを埋め込むことが可能になっている。人間がそのようなコンピュータの存在や、働きを意識することなく、そこからサービスを受けて、便利で豊かな生活がおくれるようにするコンピュータの活用形態をユビキタスコンピューティングと呼んでいる。

CPU を搭載し演算能力を有するチップや、メモリ機能のみを有するチップなどが埋め込まれた IC カードに代表されるように、携帯できる様々なツールにコンピュータを埋め込むことで、サービスを受けられる。どのようなツールにどのようなサービスを受けられるチップを埋め込むのか、あるいは、どのようなサービスがこのような形態に適しているのか、HCI の観点から検討することも必要である。

3.8 補足：電子回路によるデジタル処理の実現とその発展

これまでの議論で省略した電子工学的なコンピュータの側面について補足する。

現在のコンピュータは、電気の流れによって特定の動作を行われる電子回路によって構成されている。これまで述べてきたように、デジタル化されたデータの処理は論理演算や繰り返し処理を実現する論理回路によって行われる。そして、処理の手順がプログラムによって与えられ、プログラム自体が、論理回路によって解釈され、解釈された結果に基づき、基本的な計算の組み合わせによってさまざまな複雑な処理が可能となる。しかし、こうした処理を電子回路上への実装する、という電気電子工学の進歩がなければ、社会への普及は進まなかったであろう。簡単ではあるが、この進歩の過程について以下で解説する。

3.8.1 電子回路の登場：論理演算回路の実現

情報科学技術の発展は、デジタル化されたデータを計算するための論理回路の

高速化、高密度化と、より良い計算手順の発見、発明によって進められている。特に、論理回路の実現方法として電子回路を用いることで飛躍的な発展が可能となった。

論理回路の基本は、二つのデジタルデータを入力し、その組み合わせによって結果を出力する演算素子の組み合わせによって実現できる。電気信号で考えれば、1ビットのデータであれば、電気信号のオンとオフ、すなわち電圧の違いで表すことができるので、電圧の違いによる信号を二つ入力し、その組み合わせに応じた結果を出力する演算ゲートが組み合わせられ、論理演算が行われる。つまり電子回路では、論理演算は電気信号のオンオフを制御するスイッチの組み合わせによって実現できる。

非常に大ざっぱに言えば、このスイッチの役割を果たす素子の高速化と高密度化が、コンピュータを産み出し、その性能を向上させる鍵となってきたのである。開発されたばかりのコンピュータは、素子としてリレーが利用されたが、電磁石によって機械的にスイッチを動かしていたため、そのスイッチング速度や小型化には限界があった。また機械部品は使用頻度によってスイッチの消耗も起こることから、そのメンテナンスにも手間がかかった。

そこで機械的な部品を無くすことによって、コンピュータの性能を向上させることになった。電子的な回路のみでのスイッチの役割を果たしたのは真空管である。金属を加熱すると、金属内部で動き回っている自由電子が表面に飛び出してくることを利用して、電子の流れをコントロールするのが真空管である。その真空管にスイッチの役割をさせるようにしたのである。真空管では、電子的な機構だけで構成されることから、リレーのような機械的なスイッチと比較して飛躍的な高速化が可能となった。

しかしながら、真空管は内部構造が複雑で、真空管もガラスでできており割れやすく、動作を安定させることも困難で、結果的に信頼性も低かった。実際に、1946年にペンシルバニア大学で開発された最初のコンピュータの代表例であるENIACでは、約180,000本の真空管と1,500個のリレーで構成されており、消費電力150kW、重量30tという大規模なもので、真空管によるコンピュータは極めて特殊な機器に過ぎなかった。

3.8.2 電子回路の発展：真空管からの脱却

真空管は構成部品も多く、複雑な構造をしている。真空管を利用したままでは、回路の小型化や高密度化が困難であったが、半導体を利用した電子回路の登場が状況を一変させることになる。

半導体は、電気を通す導体としての性質と電気を通さない絶縁体の両方の性質を持った物質で、異なる性質の半導体を組み合わせることで、電子の流れをコントロールすることが可能となる。半導体を利用したトランジスタは、スイッチとしての役割を果たすことから、トランジスタを組み合わせることで、論理回路や記憶回路を作ることができる。さらに、半導体の上に、異なる性質の半導体を階層的に組み合わせることで、トランジスタと同等の回路を容易に作ることも可能となった。

その結果、複数のトランジスタを利用して作る様々な回路を一つの半導体の上にとまとめることが可能となった。このように複数の回路を1つのパッケージに集めたものを、集積回路 (IC: Integrated Circuit) という。

初期の集積回路は、ごくわずかのトランジスタを集積しただけであったが、技術の進歩によって、より微細な配線加工が半導体の上で可能になり、集積度の向上が図られ LSI (Large Scale Integration)、VLSI (Very Large Scale Integration)、ULSI (Ultra Large Scale Integration) と、半導体製造技術の進歩により、回路規模、性能が向上してきた。ムーアの法則と呼ばれるように、これまで18~24ヶ月で2倍の速度で集積度の向上が図られてきたのである。

集積回路は、簡単に言えば、半導体の基板上に微細な回路を写真の原理で定着させることによって製作される。その回路を転写する際の光学分解能をプロセスルールと呼ぶが、プロセスルールの分解能が高くなれば、より大規模な回路を作り込むことが可能となり、トランジスタの集積度が上がる。さらに、スイッチの役割をする回路間の配線も短くなることから、電流のオンとオフの閾値の電圧が下がり、より高速にスイッチングが出来ることを意味している。

つまり、集積度の向上によって、同じ長さの配線の中に、より多くの回路を組み込むことで、記憶回路で考えれば集積度の向上は大容量化を進展させ、またスイッチングが高速になったことで、処理速度も向上してきた。このような半導体を利用した電子回路を利用することで、デジタルデータを大量かつ高速に処理することが可能となったのである。

3.8.3 メモリ (記憶装置) の進化

以上は「計算」を行なうスイッチの部分の話だったが、最後に「記憶」を行なうためのメモリ (記憶領域)の電子化についても述べておこう。1949年に開発されたEDSACというコンピュータでは、メモリには水銀遅延線という水銀中の音波として情報を格納する装置が用いられた。その後、メモリには、磁気コアメモリが用いられ、現在では半導体素子であるSRAMやDRAMが用いられている。SRAMはトランジスタを組み合わせ、電源を切っても状態を保持することができる回路で

作られている。DRAM はコンデンサに電荷を保持することにより情報を格納する。RAM、DRAM は、両方とも集積回路として実現されているのである。

第4章 デジタル化・計算化の技術的影響

本章では、さまざまなデータをデジタル化し計算することによって、複雑な処理を実現するという情報科学技術の原理を踏まえ、情報をデジタル化や計算化したことの技術的な意味や価値、そして、その問題や限界について述べる。

4.1 デジタルデータの特徴

デジタルデータには、非劣化性、平等性、メディア非依存性、処理の高速性といった特徴を有している。これらの優れた特性が、情報科学技術と結びつき、大きな社会的変化を引き起こしている。

4.1.1 データの非劣化性

デジタル化されたデータは、アナログデータと異なり、0と1という二つの状態の区別をつけることが出来れば、元の状態を再現することが出来る。そのため、データの保存や複製や移動に伴い劣化することがない。デジタルデータが有するこのような特徴をデータの非劣化性と呼ぶことにする。たとえば写真を変色させずに保管するには、湿度や温度などの管理が重要である。それでも多少の変色を避けることはできない。ところがデジタル写真の場合は、デジタル化された2進列を読み出すことができれば、変化することはあり得ない。2進列は「0と1」、「磁石のNとS」、「アルミの突起と平坦な部分」のように二種類の状態の一方を書き込んだり読んだりすることで保存・通信できる。このときに二つの状態を区別する閾値の範囲であれば、元の2進値に戻すことが可能となる。

たとえば、数値が連続的に0から1へ変化している場合で閾値が0.5であれば、0の部分が錆びて0.4になり、1の部分が歪で0.6に下がったとしても、閾値の範囲内での変化であるので、読み出し可能であって、再度0と1に解釈が出来る。もちろん、閾値を超えた変化が起これば、0が0.6になってしまった場合には、区別がつかなくなる可能性がある。しかし、そのようになる前に読みだしておいて、新しい記録媒体に保存すれば、あるいは通信して送っておけば、再び0と1になるのである。つまり、2進列は、それ自体としては劣化することなく、デジタルデータは移動、複製、保管、通信することが可能なのである。

さらに、2進列となったデジタルデータには、その信頼性を高める工夫を比較的容易に取り入れることができる。すなわち、誤り検出符号や誤り訂正符号である。簡単にいえば、多少冗長なビットを付け加えることで、0が1に、あるいは1が0

に反転してしまう誤りを検出あるいは修正する方法（誤り検出符号、誤り訂正符号）である。

このように、デジタルデータを用いると、誤りや誤差が混入しにくいだけでなく、0と1という二つの状態が排他的な関係となることを活かすことで、誤りの訂正をも容易に実現できるのである。当然、現実の誤り訂正は、もっと複雑な仕組みであるが、デジタルデータの特性を活かした誤り訂正が、通信や保管する場合のデジタルデータの信頼性を高め、デジタルデータの非劣化性を実現しているのである。

4.1.2 データの平等性とメディア非依存性

アナログのデータの場合、データが持っている情報を波と見たとき、完全に相似した別の形式への変換でしか伝送できない。またその波形が少しでも異なれば、伝えられた情報は異なるものになってしまう。これは保存された場合でも同じである。そのため、アナログ情報は、それぞれの特性に応じた記録方式や伝達方式が存在し、同時に、その特性にあったメディア（すなわち伝達媒体）が存在し、その利用が必要とされるのである。

たとえば、写真の場合は、光の強さに反応する感光剤を利用することで、光の強さというアナログデータを紙の上の画像として記録する。当然であるが、感光剤には、音という空気の振動を記録することは出来ないので、画像を保存するための写真には、音声を保存することはできない。音声を記録する場合、その音（音波）の波形を電氣的に相似した波に変化させ、それをたとえばレコード盤であれば、アクリル上の溝に波を刻むことになるし、カセットテープでは、音声を電気信号へ変換し磁力の強弱によって記録する。あるいは糸電話の場合には、空気の振動を糸の振動へ変換することで音声を伝達することができる。

波形が完全に一致した場合にのみ完全にデータを記録、伝達したことになるが、現実的には、そのようなことは不可能である。多少なり、元データと異なる波形となり、アナログデータの劣化が起こるのである。

現在では、アナログデータに関してもさまざまなデータを電気信号に変換し、さらに磁気の強さに変換してアナログ電波として伝達に利用することも可能となっているが、その本質が変わったわけではない。どうしても変換に伴うデータの劣化のみならず、記録や伝送によるデータの劣化が起こってしまう。

それに対して、デジタルデータ化は、情報処理における「平等性」も産み出す。たとえば、データの圧縮処理技術においてもデータの特性を積極的に利用するのでなければ、デジタル化されたものは、文字、音、画像など区別なく扱うことが

できるのである。すなわち、デジタル化されてしまえば、どのようなデータも平等であり、その処理に関しても、同じ仕組みや手続きで行うことができる。初期のコンピュータは、デジタル化が容易な数値データしか処理できず計算機と呼ばれたが、さまざまなアナログデータをデジタル化できるようになったことで、デジタルデータを処理するコンピュータは、さまざまな処理が出来る汎用機械として、その可能性は大きく広がったのである。

さらに、デジタルデータの平等性の結果、デジタル化されたデータは、メディアに対しても非依存になる。デジタルデータを扱えるメディアでは、どのような種類のデータであってもデジタルデータとして、保管や伝達で利用することができるからである。

たとえば、CD-ROMなどのデジタルメディアには、文字・音・画像のどれを記録することも可能であり、ネットワークを利用して、これらを同一回線で通信することもできる。このように、データがデジタル化され、平等性を有していることで、同一の媒体にさまざまな情報を載せることができるようになる。この性質をメディアの非依存性という。

メディアに非依存の結果、たとえば、デジタル化されたものであれば、文字専用のCD-ROMと写真専用のCD-ROMを用意する必要もなければ、音専用の回線と絵専用の回線を用意する必要がない。メディアやネットワークが汎用に使えるのである。アナログデータが、データの種類ごとに異なる方式やメディアを使い分けなくてはならないのと大違いである。これらのデジタル化により、ネットワークの設備や仕組みは変化しないままで、文字、音声、画像、動画などをやり取りできるようになったのである。

このネットワークの汎用性は、パケット通信の技術により、さらに深まった。デジタル化された2進列は、送信時にいったん切り離し、受信後に組み立てるということも可能である。これをパケット通信と呼ぶ。パケット通信方式のおかげで、大きなデータを通信しながら、同じ回線を利用して小さなデータを通信することも可能となった。その結果、一つの回線を共有し、さまざまなデータを同時に通信可能で、効率的な通信環境も提供されるようになったのである。

以上の汎用性が、インターネット、ワイヤレスネットワーク、そして携帯電話等における大容量で低コストのネットワーク環境の普及が進む原動力となっているのである。

4.1.3 データ処理の高速性

計算世界観に基づく世界は、どんな複雑な処理であっても、それらは非常に簡単な処理を組み合わせ、その手順をプログラムとして与えることで、複雑の処理

も実現できると考えている、しかしながら、その前提として、実用的な処理速度を求めるのであれば、組み合わせられる一つ一つの処理が極めて高速である必要がある。その結果、簡単な処理を組み合わせた複雑な処理も高速で処理できるのである。

デジタルデータは、その特性から演算のルールが極めて簡単である。論理演算に関して考えてみれば、予想されるすべての入力と出力の組み合わせは限られており、非常に簡単な仕組みで高速な処理が期待できる。実際にデジタルデータの0と1の組み合わせを電気信号で表せば、スイッチの「オンとオフ」による電圧の違いによって表わすことができ、スイッチの動作によって論理演算が実現可能である。さらに、スイッチの組み合わせによって、さまざまな論理素子を構成することができる。こうした電気信号を制御するスイッチを半導体によって論理素子として構成し、複雑な処理を行う電子回路として、高速なデジタルデータの処理を可能としたのがコンピュータである。

高速化のもう一つの大きな要因は「自動化」である。コンピュータは、プログラムさえ定めておけば、プログラムで示された手順に従い、デジタルデータを人間では不可能な速さで処理することが可能である。この自動化により、膨大な情報の収集、加工、発信が可能となったのである。

高速性は進歩の速度にも現れている。ハードウェアを根本的に変革する新しい技術の創出には、相応の時間を必要とする。それに対し、ソフトウェアの改良による新しい技術への進展は速い。その理由の一つは、多くの技術が、つねに汎用性を考慮して開発されており、多様な組み合わせの可能性が残されていることにある。実用に供している個々の技術は、その機能や役割が明確である。アイデア次第でそのような技術を組み合わせることが可能であり、結果として個々の特徴が生かされた新しい技術へと変化することがある。

4.1.4 デジタルデータの恩恵 :いくつかの例

以上あげてきたデジタルデータの特徴 :非劣化性、平等性、メディアの非依存性、そして高速な処理が可能であるということが、具体的にどのような意義を持つのかという点について、実用例を簡単に紹介しよう。

デジタルデータを利用するデジタル通信は、それまでのアナログ通信だけでは考えられない様々なサービスを実現するようになった。これは、単に通信を効率化しただけではなく、コンピュータなどの処理装置と他の装置を、通信によってシームレスに接続することで初めて実現できたことである。情報ネットワークの出現である。このような大掛かりな仕組みが実用化されたのは、それがほとんど

すべての用途に使えたからである。情報ネットワークの整備がサービスに依存しなかったためである。たとえば、インターネットは、デジタル化したデータのやりとりをするネットワークであるが、デジタル化してしまえば、テキストであろうが、数値であろうが、画像や音声、そして動画でも自由にやりとりできる。だからインターネットを整備しよう、という動きが出たのである。

こうした情報ネットワークは、場所の問題に対する解決も導き出しつつある。一種のユビキタス性の実現である。たとえば、以前なら銀行に直接出かけ、現金を引き出して買いものを行なう必要があった。しかし情報ネットワークの出現により、クレジットカードと銀行口座の残高があれば、ネットワークを利用して簡単に店舗やネットショッピングサイトで買いものを行うことができる。また、たとえば伴奏付で歌を唄う場合も、実際の演奏家を部屋に呼ばなくともよい。通信カラオケは楽譜データをセンターから各店の演奏機に転送し、演奏機が楽譜を音にしてしまうのである。

データ処理の高速化は当然様々なところで役に立っているが、高速化が実現の鍵となった例を示そう。医療分野で利用されている CT スキャンである。これはコンピュータの登場によって可能となった代表的な技術である。

CT スキャンは、検査対象の周囲を線源とセンサが回転し、検査対象は X 線を全方位から受ける。照射された X 線は検査対象を通過し、対象に一部吸収されて減衰した後、線源の反対側に位置する X 線検出装置に到達し、それぞれの方向で、どの程度 X 線が吸収されたかを記録する。そこまでがスキャンの部分である。

これは原理的にはレントゲン (X 線写真) と変わらない。普通の X 線写真は、一つの方向からの X 線の強さを写真の原理で 2 次元の画像化したものである。一方、CT スキャンでは、画像にするために、各方向からの X 線の強さのデータから 1 断面を格子状に分割し、各部位の吸収率を未知数とし、その合計が実際の吸収量と等しくなるように巨大な連立方程式を立て、それを解くのである。人間ではとてもできない。データをコンピュータで処理することを前提として、考え出された技術なのである。

画像を構成するのにかかる時間はその処理速度に大きく依存している。実際、CT の実用化当初は撮影してから画像が出力されるまでに長い待ち時間を要していた。しかし、数理科学の手法を用いたアルゴリズムの進歩により、現在は X 線照射とほぼ同時にリアルタイムで画像を確認できるまでに高速化している。

ここで高速化の重要性について、もう少し具体的に示しておこう。よりよい精度を得るためには解像度の向上が重要だが、そのためには処理速度の改善が必須である。たとえば通常は一つの断面を 512×512 に分割しているが、解像度を高めるために各行列 2 倍の 1024×1024 に分割すると、データ量は 4 倍 ($=2 \times 2$) になり、

計算量はそれ以上になる。つまり、「2倍の良さ」を実現するためには「4倍(以上)の速さ」が必要となるのである。このような状況は、様々な応用技術で生じている。つまり、規模や精度を少し上げるだけでも、非常に大きな高速化が必要となってくる場合が少なくない。コンピュータが発達し、情報機器への依存度が増すにつれ、ますますの高速化が必要となってくるのである。

4.2 デジタル化と計算化の問題点

これまでは、デジタル化と計算化がもたらす技術的な「恩恵」の部分について主に述べてきた。ただしそれだけではない。デジタル化や計算化が、逆に技術的に困った問題を産み出す元になっている場合もある。この節では、そうした技術的な問題点について述べよう。我々は技術的な問題点をも理解した上で、情報科学技術を利用していくべきである。

4.2.1 デジタルデータは近似にしか過ぎない

デジタルデータでは、非劣化性を有しているのがデジタルデータとして、通信や保管の際にデータの劣化は起こらない。しかしながら、デジタルデータへの変換の際に、アナログのデータは、必ず近似したデジタルのデータにしか変換できない。デジタルデータの段階では、さまざまな処理の過程において劣化は起こらないが、デジタルデータへの変換の際に、必ず劣化が起こるのである。その劣化をどの程度まで許すのか、これは技術とコストのトレードオフである。

デジタルデータは離散値であるので、連続的なアナログデータは近似した離散値に変換する必要がある。デジタルデータの2進列を多くすればするほど、より良い近似が可能となるが、処理すべきデジタルデータも増えることから、記憶する領域や処理量も増えることになる。

たとえば、アナログデータである音(波)をデジタル化する際には、階段状の棒グラフによって近似することになるが(2.1節参照)、その刻み幅と波の高さ(振幅)の段階の程度によって、近似度合いが異なることになる。

CDの規格では、刻み幅は44.1kHz(1秒間に44,100回)、波の高さは16ビット(65,536段階)でデジタル化されている。当然、現実には滑らかな曲線として表現される波が、棒グラフによって離散的に表現されているので、本来の連続的な波が階段状に表現されることになり誤差が発生している。

もちろん、デジタル化の過程で、より精度を上げることも可能である。たとえば、刻み幅をより細かく、また波の高さの段階をより細かくすることで、元の波により近づけることが可能となる。しかしながら、その分処理すべきデータ量が増えることから、処理量も増え、必要となる記憶領域も大きくなる。つまり、よ

り高速な処理と記憶容量が必要となり、その時点での処理技術レベルの制約を受けることになる。

CD が規格化された当時の技術水準では、音をリアルタイムにデジタル化する、あるいは逆にデジタルデータをアナログデータに変換し、我々の聞くことが出来る音にリアルタイムに変換し再生することができる限界が、刻み幅は 44.1kHz、波の高さは 16 ビットであった。その後の技術進歩によって、今では、たとえば、刻み幅 192kHz、波の高さを 32 ビットで変換することも実用的に利用されるようになってきている。この場合、CD と比較すると、刻み幅が 4 倍、高さが 2 倍のデータ量が必要となるので、処理すべきデジタルデータは 8 倍となる。つまり、同じ時間で処理をするのに 8 倍の処理速度と 8 倍の記憶容量が必要になる。しかし、それでも完全にオリジナルと同じ波を再現できる訳ではなく、あくまでも近似したものにしかならない。デジタルデータへ変換する際に、原理的にはビット数を増やすことによって精度をいくらかでも高めることは可能であるが、技術の水準に応じ、実用上問題がないレベルに設定されることになる。

4.2.2 数値計算には誤差が発生する

原理的には計算には誤差は発生しない。もちろん、無限桁の 2 進列の計算はできないので、無理数や無限小数を表わすことは不可能である。たとえば、分数 $1/3$ を $0.333\dots$ という形では表わすことも、扱うこともできない。これを仕方なく有限の桁数で表わそうとすると、当然、誤差が生じる。しかし、有限桁の 2 進列として表わされたものに対しては、その演算で誤差が生じることはない。

というのはあくまで原理上の話である。現在のコンピュータでは、ある固定した有限のビット数 (桁数) で数値を表現しているため、その精度や範囲にも限界がある。そのため誤差が生じるのである。

固定した桁数では表現できる数値の最大値と最小値が存在する。最大値を超えた場合はオーバーフローという。小数においては、絶対値がある値以下の数は 0 とみなされるが、これをアンダーフローという。まずは、このように扱える数値の範囲の制限が存在する。

デジタルデータ同士の計算の過程において発生する誤差としては、「丸め誤差」、「打ち切り誤差」、「情報落ち」、「桁落ち」がある。誤差の原因は、データ形式とも密接に関係しているので、誤差の原因が分かれば、計算の順序を工夫することで、誤差を少なくすることも可能であり、また誤差の範囲も予測できることから、現実には必要な精度を、もちろん、桁数を限らない表現法や、そうしたデータに対する計算も可能ではあるが、それでは計算コストがかかってしまう。有限桁に限ることで、その範囲までの処理を回路化することで、通常のコンピュータは高速

化を実現しているのである。たとえば、32ビット演算回路では、扱える整数は約21億、正確には $2^{31} - 1$ まで確保することが可能である。

丸め誤差とは、切り捨てや四捨五入によって生じる誤差のことである。たとえば、正しい値が7.89であるデータを小数点以下で切り捨てれば7となり、小数点以下2けた目で四捨五入すれば7.9となり、それぞれ誤差が生じる。ここで「誤差」とは、正しい値と実際の値との差のことで、7.89と7.9の誤差は+0.01、7.89と7の誤差は、-0.89である。

丸め誤差の原因は、コンピュータが取り扱うデータのビット数(桁数)が有限なことに起因する。桁数をはみ出したデータは、本来なら切り捨てられてしまうか、あるいは四捨五入されてしまう。そのときのはみ出したデータ分が誤差となるのである。有限な桁数で表現できる数値も存在するが、すべての数値が有限の桁で表現できるわけではない。

打ち切り誤差とは、特定の桁でデータが打ち切られてしまうことで生じる誤差のことである。たとえば、 $1 \div 3$ の計算結果のような循環小数や、2の平方根の値のような無理数(無限小数)をコンピュータで取り扱おうとすると、必然的に打ち切り誤差が生じてしまう。

情報落ちとは、浮動小数点数の計算において生じる誤差である。浮動小数点数の計算では、指数を揃える必要が発生するので一般に大きい値の指数に合わせるが、絶対値の大きい値と小さい値の足し算や引き算では、小さい値の方が仮数部が雑な値になり情報が欠落してしまう。このときに発生する誤差が情報落ちである。

たとえば、仮数部の表現がビットに限られていた場合に12と20の足し算を考えてみる。計算では指数をまず合わせるが、たとえば $12 = 1.2 \times 10^1$ と $20 = 2 \times 10^1$ と合わせると、仮数部は整数なので1.2は1と切り捨てられて、和が $(1 + 2) \times 10^1 = 30$ となってしまう(正解は32なのに)。これが情報落ちである。

桁落ちとは、浮動小数点の計算において絶対値のほぼ等しい二つの数の引き算を行う場合に、有効数字が減少することで、数値の信頼度が下がることである。一般に、科学計算の世界では、有効桁数を揃えるということが厳しく要求されるので、たとえば、2と2.000は意味が異なる。2は極端に言えば、1.5から2.4まで

用語解説6：浮動小数点

小数を表現するために、指数を利用して $0.25 = 25 \times 10^{-2}$ と書けることを利用し、小数を二つの数(たとえば(25、-2))で表現すること。非常に小さな数から大きな数までを表現する方法である。一般に (x, y) のうち、先の例の25にあたる x が仮数部、-2にあたる y が指数部という。

普通に使う十進法の場合、指数の基数は10となる。つまり、指数部の値 y は $\times 10^y$ を意味する。指数部に符号の情報をつけることで、小さい数字から大きい数字まで表わすことができる

の範囲の可能性があるのに対して、2.000 では 1.9995 から 2.0004 までの範囲となり、その数値の意味が全く異なるのである。

浮動小数点の計算の場合、二つの数の引き算の結果、上位の桁が 0 になると自動的に指数の桁が一つ上がるため、有効桁が 1 桁減ってしまうことになる。その結果、下の桁の数値が信頼できないものになる桁落ちが発生してしまうのである。

たとえば、 $1.02 - 1.01 = 0.01$ であるが、浮動小数点の計算では、 $1.02 = 102 \times 10^{-2}$ 、 $1.01 = 101 \times 10^{-2}$ と、仮数部は 3 桁の精度があるのに対し、答えは自動的に $0.01 = 1 \times 10^{-2}$ に変換されるため、仮数部が 1 桁だけに落ちてしまうのである。

これらの誤差は、あらかじめ固定された桁数の 2 進数で数値を表現することに起因するもので、コンピュータで（ある範囲までは）高速に処理したい以上、この手法から逃れることはできないだろう。したがって、コンピュータ上でデジタル化された数値の計算では、ある意味で誤差は必然なのである。そのことを理解し、誤差の発生を少しでも少なくするための計算上の工夫を行うと同時に、誤差の範囲を見極め、必要な精度を確保するために、利用する桁数などを選択する必要がある。

4.2.3 計算には間違いが入り込む可能性がある

デジタルデータを処理する計算化では、非常に複雑な処理も簡単な処理の組み合わせでできることを述べた。こうした計算は、電子回路での演算の組み合わせとして実現できる。電子回路の信頼性は高く、一つ一つの処理に関して間違いが起こることはほとんどあり得ない。しかしながら、それらの処理の順序や手順を指示するプログラムに間違いがあった場合、処理結果も間違えた結果しか得られない。現実問題として、そのようなプログラムの間違いは頻繁に起きる。簡単な処理の組み合わせで複雑な仕事を表わすのだから間違いが起きるのも当然だろう。このようなプログラムの間違いのことを（プログラムの）バグ (bug) という。

バグは、さまざまな段階で入り込む可能性があり、完全になくすことは困難である。そして、その影響は極めて大きくなりつつある。コンピュータは社会に浸透し、その重要性も増しているが、このようなバグによる間違いによって、コンピュータが想定外の動きをすることがありえる。コンピュータが社会システムの中で重要な役割を果たすにつれて、このような間違いが発生した場合に影響も大きくなっている。

バグにもいくつかレベルがある。(1) 処理全般の考えかたや手順に間違いがあり、仕様やアルゴリズムのレベルで間違いがあって、結果として計算の結果が異なる場合（論理的な構造によるもの）。(2) アルゴリズムは正しいが、実際のプログラムとして実現する過程で間違いが入ってしまう場合（コーディングエラーによるもの）。(3) プログラムを実際にコンピュータに入力する段階で間違えてしまう場

合。さらには、(4) プログラムには間違いはないが、入力されるデジタルデータが仕様と異なったり、思わぬ誤差の影響で、本来の処理と異なる結果になる場合などがある。さらに、当初は問題なかったとしても、環境の変化によってバグとして扱われるようになるものもある。たとえば以下に紹介する 2000 年問題がそのよい例である。

このように、単純なものから、様々な要因が複雑に絡んで発生するバグまで、計算には何らかの間違いが混入する可能性は極めて高い。情報処理の様々な手法は、その汎用性を高めることにより、プログラムというソフトウェア面で様々な処理を可能にしてきた。また改良が容易なようにソフトウェアを設計してきた。こうした点が、バグの混入を許す原因となったのである。

バグに対する対策も初期の段階から行なわれてきた。その結果、コーディングエラーを起こしにくいようなプログラム作成環境が提案されたり、仕様に対してプログラムが正しく動いているかをテストする手法や、正しさを検証する手法が多く提案されてきた。しかし、高度なバグを排斥するのは困難であり、完全に間違いがなくなることはあり得ない、という点に注意が必要である。

情報技術が社会のさまざまなところで利用され、社会基盤として重要な役割を果たすようになってきた以上、このようなバグの存在で、間違いが発生した場合の社会的な影響も大きくなっている。特に、情報ネットワークによって、様々な情報システムが相互に接続され、多種多様のデータのやりとりが可能となったことで、様々なサービスが実現し便利になっているが、逆に個々の情報システムの影響が他のシステムにも及ぶ可能性も高まっており、結果として社会基盤における情報システムの脆弱性も高まっているのである。

例 11 : 2000 年問題

西暦 2000 年に入ったときに、多くのコンピュータプログラムに異常が生じる可能性がある、と指摘された問題。本来年号で4桁確保すべきところを、省略して下2桁のみを利用するように作成されていたプログラムが多く、2001年と 1901年が 01年のままで区別がつかず、結果として計算がおかしくなってしまう、という問題である。

プログラム自体が 2000年を過ぎてまで利用されることは想定しておらず、それまで問題がなかったものが、2000年以降も、つまり本来予測されて耐用年数を超えて運用されてしまうことでバグとして表出することになったものである。環境に依存してバグとなってしまった例であろう。

情報学リテラシーはなぜ必要なのか？

ユーザの立場から

コラム No. 3

様々な科学技術領域の中で、特に「情報」が他と異なる特徴として、領域自体が 20 世紀半ばまでは存在しなかったこと、かつ、その出現によって文字通り社会が大きく変化したことが挙げられる。その結果、現時点の社会成員の中に明らかなリテラシーの有無の差が存在し、それに伴う得失も現象として見えやすくなっている。

多様な IT 機器が「高齢者にとって使いにくい」原因については、現時点での高齢者 (65 歳以上) では「情報という概念」がない、あるいは希薄であるという要因が、他の要因 (脳神経系の高齢化に伴う認知的機能低下、および社会文化的背景下での態度・方略) とあいまって、個別機器の利用に必要な学習を阻害していることを指摘した研究もある。文字通り、情報リテラシーの欠如によって「機器が使えない」という問題が発生しているのである。

IT 機器が使えないことが「生きていく」上の問題となるだろうか。「使わなくても生活はできる、全員が必ず使わなければいけない必然性はない」という主張はしばしば聞かれるところであるが、現実には社会の情報化が他の要因 (社会経済活動の高密度化、人的コストの削減化等) の影響下で、日常生活の細部にまで行きわたり、「使わないことが実際上のデメリット」となりつつある。たとえば携帯電話が広範に普及した結果、使わない人は、公衆電話の減少による不便さ・不安、また「携帯で連絡が取れない人には情報がいきわたりにくくなる」といった活動上の制約を被っている。

さらに社会にいわゆるユビキタス・システムが広まる中、自ら主体的に IT 機器を使うことを選択していないにもかかわらず、自分の身体・活動・存在が情報処理の対象となってきた。たとえば交通機関の料金の IC カード化や公共空間の監視カメラモニタリングなどは、単にその時点での処理だけでなく、ログ情報のマイニングなどによって「様々な個人にかかわる情報が処理される」可能性を持つ。また家庭への各種「見守りシステム」の導入が、見守られる側と見守る側の間に権力関係を持ち込み、強化していく可能性も示されている。

これらはデジタル・デバイドの問題とされるが、世代間のみならず、地域・社会階層・性別などにも存在する。情報世界にどれだけ触れ、その中で主体的に活動する機会があるかということと、このリテラシーの獲得には強い関係があると考えられ、現在の一過性の問題としてではなく、持続的問題としてとらえていく必要がある。こうした問題や悪用可能性から「自分や家族を守る」ためにも情報リテラシーは必須となってきた。

こうした現状は、科学技術を推進する側にも「利用者側から科学技術を見るリテラシー」が必要とされる現状を示唆しているようにも思われるが、いかがであろうか。

第5章 デジタル化・計算化の社会的な影響

本章では、情報科学技術の社会的な影響について、情報科学技術のどのような特質が、その原因になっているのか、という点を中心に、事例紹介の形で述べる。まず一般的に、電子的に処理される過程が「見えない」ということがもたらす影響を述べ、その後で、具体例として、ICカードシステム、インターネットによる知の形成、そしてネットオークションの話をし、最後に情報格差（デジタルデバイド）について述べる。

5.1 見えないデータ・見えないコンピュータ

情報科学技術を利用したコンピュータやネットワークには、「データは見えない」、「データは物ではない」、「コンピュータは隠される」という特徴がある。これらの特徴がどのような影響を与えるのかを以下で議論していく。

5.1.1 見えないデータ

情報を文字や記号、数値などで表わしたデジタルデータについて考えよう。データは物ではない。それゆえに、データを大量に複製・送信することが可能となる。よい情報や役に立つ情報だからといって、それを複製することが「よい行為」であるとは限らない。他者が知的所有権を有するデータの複製・公衆送信は権利侵害となる。個人情報や、人を不愉快にさせる情報の複製・公衆送信も「よい行為」に該当しない。

データは見えない。つまり、データが転送されたり保存されている過程は見えない。それらが画面などで外に表示されてはじめて見えるのである。そのため、データの好ましくない複製や公衆送信が行なわれていることは、コンピュータやネットワークを眺めていてもわからない。

「見えない」ことの顕著な例は迷惑メールだろう。迷惑メールは、大量に複製され、広く配布されている。紙に印刷された郵便物を100万通送

用語解説7：迷惑メール

不特定多数に送られる（不必要な）情報を、一般に迷惑メールという。公開されているウェブページに書かれているメールアドレスを自動収集するなどして集めたアドレスに対して、広告等を送る。ある程度の機器さえあれば、低コストで多量の宛先に送ることができるのである。

なお、迷惑メールは俗にスパム（spam）ともいうが、まったく関係のない SPAM という商品名に配慮して（とくに日本語では大文字、小文字の区別が付けられないので）公には迷惑メールと呼ばれている。

信するには多額の費用と手間がかかるが、100万通の迷惑メールを送信するのはそれに比べればはるかに簡単である。これは、作成と配送が「見えない」ということに等しい。チェーンメールも同じように、見えない状態のまま大量に複製され、配送され、メールシステムを占有してしまう。

たとえば、暗い部屋で電灯をつけると、部屋は明るくなる。自動車で急な坂道を登ろうとすれば、エンジンの回転数は高くなる。このように、多くの器具・機械は、その動きが外部からすぐにわかる。だが、大量のデータを処理しているコンピュータは、そうでないコンピュータと比べて汗をかいているというわけではない。ネットワークを大量にデータが流れても、ケーブルの色が変わるわけでもない。つまり、人間の目では、データの流れを簡単に把握することができない。その結果、大量データの複製や公衆送信が行なわれていても、それに利用者は全く気がつかないまま過ごしてしまうのである。

5.1.2 隠されるコンピュータ

現代社会では、データのみならず、コンピュータも見えないように隠されつつある。たとえば、昔の自動車では、エンジンの回転数を制御するキャブレターは、アクセルペダルと金属ケーブルでつながれ、燃料の霧吹き具合を制御していたが、それでは微妙な燃料噴射量を制御することができず、不完全燃焼により有害ガスを発生させてしまう。現在の自動車は、1秒間に百回以上の燃料爆発のすべてについて、コンピュータが適切な燃料噴射量を計算している。このように、機械に組み込まれたコンピュータからなるシステムが組み込みシステムである。利用者の多くは、このような緻密かつ高速な計算が自動車のエンジンに必要なこと、そのおかげで排出ガスによる公害が減少していることを知らない。

同じように、交通機関では、あらゆる場所にコンピュータが使用されている。銀行や証券会社でも、取引にコンピュータは欠かせないものになっている。このように、様々な複数のコンピュータや機器を組み合わせで作られる仕組みが情報システムであるが、自動車の場合と同じように、これらの取引をコンピュータがどのように支えているのかを理解している利用者は少ない。

組み込みシステムも、情報システムも、コンピュータが非常に重要な役割を演じているにもかかわらず、コンピュータは表に出ないように隠されている。それはデータが物ではないからこそ可能な隠蔽である。しかも、利用者の多くは、コンピュータと無関係だった従来の機器、従来の人的システムが、まだ存在し続けていると感じている。それは、データもコンピュータも見えないように隠されているからに他ならない。

このように、現代社会の複雑な生産、流通、消費などの経済活動や、人間相互のコミュニケーションなどの文化活動のほとんどが、コンピュータ・ネットワークという情報基盤の上に成立している。それにもかかわらず、それが多くの人々には見えない (見せない) ようになっている。時々発生するコンピュータ・ネットワークの事故・故障が、私たちの社会生活に大きな影響を与えるたびに、我々はコンピュータの存在を思い出すだけなのである。多くの利用者は、ほどなくその存在を忘れてしまう。

問題は、利用者だけでなく、情報システムの企画や運営を指揮する立場にいる人たちまでもが、コンピュータとネットワークの存在を忘れてしまう場合がある点だ。

そしてまた事故・故障の発生が繰り返されているのである。

5.2 ICカードシステム

情報科学技術の成果の一つに、IC カードを利用したシステムの構築と普及がある。IC カードが現実的な価格で販売されるようになると、現代社会の隅々に IC カードが組み込まれるようになってきた。「計算するカード」が、社会をどのように変えているのか (変えつつあるのか) について述べる。

5.2.1 ICカードの仕組み

IC カードとは、超小型のメモリを内蔵し計算機能を有する回路を持つ IC (これを IC タグという) を埋め込んだカードであり、その認証機能を用いて、偽造を防ぐ必要がある紙幣・貨幣・有価証券・身分明証などに利用されている。

ほとんどの IC カードは電池を内蔵していない。利用者が IC カードをカードリーダーに近づけると、微小電波とカードの内部に設置されたコイルを利用して発電された電流で、IC カードは動作する。IC カードは電波と内部メモリの内容を読み出し、その内容にしたがって計算を行ない、計算結果をカードリーダーに送り返したり内部メモリの内容を書き換える。電波を利用することから、非接触で情報の読み取り・書き取りが可能なのである。なお、電源がなくてもメモリの内容は保持される。

以上は IC カードの装置としての仕組みであるが、IC カードの利用において、情報科学技術的に重要な点は個人認証の導入である、技術的には公開鍵暗号によるデジタル署名 (第3章参照) に基づいた認証である。

多くの IC カードシステムでは、このデジタル署名による認証を採用している。つまり、IC に秘密鍵を記憶しておき、利用時に、その秘密鍵で (各個人の公開鍵に対応する) 署名を計算し、IC カードから相手に電子署名を送る、という方式を採

用している。これによって正当な利用か否かの確認ができるようになっているのである。

5.2.2 ICカードの役割の変化：偽造防止 → 情報収集

ICカードの役割が変化しつつある。本来は本人確認や偽造防止のための手段であったのだが、それがデータ収集の手段へと大きく変わりつつある。このような利用法の大変革が、いつのまにか起きてしまうのが情報科学技術のすごいところであり、また怖いところでもある。

紙幣・貨幣・有価証券・身分証明証などは偽造の脅威にさらされている。このようなことを防ぐために、本人の認証が確実にできるICカードが導入されたといったらよいだろう。ところで、偽造は発覚しなければ事件とはならない。過去に多くの偽造事件が起こっていたかもしれないが、我々はそれに気付いていなかったのかもしれない。偽造の防止には、発見だけでなく、発見の事実の情報収集が重要だったのである。

ICカードシステム導入の目的は、偽造等の発見だけでなく、こうした情報収集の効率化でもあった。ところが、最近では、その情報収集が別の用途に使われるようになってきた。各個人が、いつどこで買い物をしたか、いつ電車に乗ってどこからどこまで行ったか、などの情報が収集管理できるようになってきたのである。こうした情報は企業や行政にとって非常に有用な情報になり得る。さらに企業や行政が、こうした情報を上手に使うと、よりよいサービスを行えば、それはユーザにとっても嬉しい話である。しかし一方で、プライバシー侵害の問題が起きる可能性もある。

解説8：ICタグの利用

ICタグは非常に小さい。その点を生かして、ICカード以外でも、ICタグは様々な情報収集・情報管理に用いられるようになりつつある。

たとえば、スーパーマーケット内の全商品にICタグが取り付けられていれば、そのデータを利用することで、入荷数の確認、値札の自動掲示、レジでの自動精算など、商品管理を簡単に行なうことができる。さらに、商品価格の自動掲示、レジでの自動精算などもICタグの情報を読みとることで可能となる。また、各商品に付けられたICタグを利用する家庭用冷蔵庫が普及すれば、商品を購入した人は、賞味期限を把握することができる。電子レンジであれば適切な調理方法を示唆できる。

人間や動物、自動車などの動く対象にICタグをつけることで、それが今どこにあるのか。どこに向かっているのかを、把握することができる。実際、「子どもの居場所確認システム」の利用がある。また、絶滅危惧種の動物の生態を研究する際にも用いられている。

個人が「他人に知られたくない（情報の）領域」がプライバシー領域である。この領域には、個人に関わるさまざまな情報・データ（個人情報）が存在する。情報科学技術の普及以前は、個人情報を他者が入手する手段も規模も限られていた。しかし IC カードを利用すると、プライバシー領域にまで入り込んだ様々な個人情報を正確に、非接触で収集することが可能となったのである。

たとえ技術的には可能でも、プライバシー領域を侵害する行為は禁止されるべきである。そこで、日本では個人情報保護法が制定され、「本人に通知しない個人情報の収集」、「収集した個人情報の無断転用」、「収集した個人情報の無断外部提供」などが違法行為とされるようになった。

情報科学技術の普及は、プライバシーの問題を顕在化させ、個人情報の利用問題について法的対応を制定させることになったのである。

5.3 新しい集合知

爆発的な発展をとげたインターネットは、社会の構造にも影響を及ぼすようになってきている。その一つがインターネット上の集合知である。集合知とは、多数の人間で構成される社会がつくり出した知性である。この集合知のあり方にも大きな変化が生じている。

5.3.1 新しい集合知の形成

情報科学技術が登場し普及する前から集合知は存在していた。たとえば、同じ街の住民同士は、その街が直面している問題に対して多様な意見をもっている。街を擬人化してとらえると、あたかも街が一人の人間のように振舞い、街が知識を保有しているように見える。これが、その街のもつ集合知である。企業の場合でも同じである。企業が一人の人間のように振る舞っているように見える。街の集合知は、住民が街全体として問題に取り組んだ結果として形成されるものである。企業の集合知は、従業員が企業全体として問題に取り組んだ結果として形成されるものである。街の場合も企業の場合も、構成員同士が頻繁に情報交換を行なうことか前提である。したがって、集合知の形成には、住民・従業員相互のコミュニケーションが深く関わっていた。

だが、情報科学技術がもつ情報のデジタル化という性質のおかげで、データを簡単かつ大量に複製し、それを遠隔地に届けることができるようになった。そこで人間は、自らがもつ知識や考え方をデータとして表現するばかりでなく、それを簡単に多くの人に見せることができるようになった。

この結果として現れたのが、インターネット⁵上で形成された集合知である。たとえば、ウェブ上の百科事典「ウィキペディア」などである。この「新しい集合知」は、多数の人間で構成される社会がつくり出した知性であるが、それまでの社会では作り得なかった性質のものである。

集合知を形成するためには、構成員が相互にコミュニケーションをとる必要がある。そのため、従来の集合知の対象は、地域や企業といった場所に根ざした集団に関わるものに限定されていた。ところが、インターネットを利用することで、知識の共有や流通が容易になった。その結果、街や企業のように顔と顔を会わせてのコミュニケーションが成立する場所のみならず、その他の場所でも集合知を形成することが可能となった。そして、先述の対象とは異なる集団が形成されるようになり、場所に根ざしていない集団による集合知が発生するようになった。このことが、情報科学技術によって形成される集合知の最も大きな特徴である。さらに、ソーシャルネットワーキング・システムの普及に伴い、同じ料理を好きな人、同じ思想の持ち主、同じ癖の人などが、情報ネットワークを利用してコミュニケーションを行ない、集合知を形成するという、従来ではありえなかった現象が現実のものになっている。

例12：インターネット上の集合知

従来、お互いを知らぬ同じ作家のファン同士が相互に作品について意見を述べ合うためには、「ファンクラブ」のような組織が必要だった。特定の作者に限らず、多くの作品に対して感じた意見を述べることができたのは、プロフェッショナルな文芸評論家に限られていた。

しかし、現在では、そのような組織に属さなくても、プロの評論家でなくても、事実上誰もが「ある作家に関するウェブ掲示板」を利用することが可能となった。従来なら文芸雑誌や新聞などの読者投稿欄にさえ取り上げられなかったような意見や知識さえも、インターネットを利用して、多くの人に伝えることが可能となったのである。

またウェブを利用した百科事典「ウィキペディア」のように、限られた専門家が百科事典を作るのではなく、インターネットを通じて多くの人々が知識を出し合い、それを共有し合うことにより成り立つ百科事典も登場している。

情報科学技術が取り扱う対象は文字だけではない。音や映像も対象である。それゆえに、同じ音楽や動画を好む人同士のコミュニケーションの結果として発生した集合知や、アマチュアの音楽製作者・映画製作者の登場を可能にし、それは、プロフェッショナルの作品にも大きな影響を与えつつある。

⁵ インターネットは情報通信ネットワークの形態の一つにすぎない。本来は「情報科学技術によってもたらされた情報通信ネットワーク」と言うべきだが、ここでは世間でよく使われている呼び名に従って「インターネット」と呼ぶことにする。

5.3.2 情報科学技術による集合知の光と影

インターネット上の「新たな集合知」の形成は、従来では取り上げられなかったような小さな文化・文明の存在を多くの人に気付かせることになった。言い換えるなら、多様な文化・文明は、インターネットによって形成されたのではなく、インターネットによって、その存在が「集合知という公共知」になったともいえる。

たとえば、ウェブを利用した百科事典「ウィキペディア」には、従来の百科事典では決して掲載されないような項目が多数掲載されている。あるラジオ番組の名前を項目名として調べると、番組の歴史、その番組を担当していたナレーターの変遷、スポンサー企業の変遷、主なゲストなどが並んでいる。そこで「ウィキペディア」を利用してスポンサー企業を調べると、その企業の主な商品、歴史、経営状況などが簡単にわかる。ラジオ番組のナレーターについての集合知、企業についての集合知、いずれも、インターネットの普及によって初めて集積できるようになった知識である。

ところが、新しい集合知の登場は、新たな問題を引き起こしている。ページ数上限の都合で紙媒体の事典には掲載されなかったような些細な知識を掲載することが可能となった。また、誤った情報の掲載も可能となったのである。従来は、多くの人目に触れる内容は、プロフェッショナルである編集者・ディレクターによって内容の取捨選択・編集などのチェックを経て初めて公衆に掲示されてきたが、新しい集合知にはチェック機能がなく、正しくない内容や判断が分かれること（不確定なこと）までもが掲示されるようになったのである。その結果、重要な項目の正しい記載と、些細な知識、誤った内容の記載が混在してしまった。

どの項目が重要で正しいのかを判断するのは、情報の受け手側の責任となった。新しい集合知は、私たちに、知識の重要性や信憑性の評価能力を求めることになったのである。

5.4 ネットオークション

ネットオークションは、ウェブを利用したオークション管理システムであり、商品掲示、質疑、入札、応札、開札、落札などを管理する。これらのほとんどの行為は情報科学技術を利用している。

5.4.1 従来のオークション・入札・中古品流通・個人売買

ネットオークションの仕組みが普及する以前は、中古品の流通、個人売買、競争入札のそれぞれが、それぞれに固有の仕組みを利用していた。

中古車や中古住宅、美術品などの流通においては、業者同士がオークションを利用して商品を仕入れ、それを一般の顧客に販売する。顧客への販売価格は販売

業者が算定し、通常の顧客は価格の決定権を持たないが、売買交渉において値引き要求などを行なうこともあった。商行為は店舗やセールス先で行なわれた。

個人売買の場合は、販売する側も購入する側も個人であり、販売する側は仕入れ価格などに左右されずに販売を行っていた。個人売買を行ないたい場合は、街頭や職場の掲示板などを利用して商品説明と希望価格を掲示したり、専門雑誌や地域コミュニティ誌を利用して広告を行なった。

競争入札は、自治体や企業が行なうものであるが、オークションとは逆に、商品やサービスを提供する側が応札者となり、もっとも安い金額を入れたものが落札者となる。

競争入札や落札は、通常は、その自治体や企業の建物内で開催される。

以上の商業行為は、ウェブを用いてインターネット上で運営されているネットオークションによって統合されつつある。

5.4.2 ネットオークションの実例

ネットオークションは、(1) 内容掲示、(2) 希望価格の入力、(3) 落札者の決定、(4) 決済、という手順で進む。そこには（ネットワーク上の）仲介者が存在する。そのやり方を簡単に紹介しよう。

- (1) 内容掲示：ネットオークションでは、出品者が商品を掲示するところから始まる。商品掲示の際には、デジタルカメラなどで撮影された商品画像、商品のメーカーサイトに掲示されている仕様や説明書などへのリンクを使用する。
- (2) 希望価格の入力（応札）：応札はウェブを通して行なわれる。なお、応札予定者が出品者に対して質問を行なう場合があるが、これは他にも公開されるようウェブ掲示板を利用することが多い。「競り」の要素が入る場合には、それまでに行なわれた応札者の金額などが表示され、応札者が額を再度設定することができる。
- (3) 落札者の決定：オークション期間を終了すると、仲介者が落札者を定め、出品者と落札者に対してオークションが終了したというメールを送信する
- (4) 決済：品物と代金の授受は宅配便を用いることが多いが、送金はネットバンクと呼ばれるウェブ対応の銀行を利用することもある。

このように、ネットオークションが普及した結果、中古品の流通、個人売買、競争入札が統合されつつある。いずれの行為も抽象化すれば、上記の手順に当てはまり、同じ行為モデルのもとに実現可能だからである。結果として、中古品販売業者も、個人も、自治体も、ウェブを利用する場合は同じシステムを利用することが増えてきた。

5.4.3 ネットオークションによって変わった世界

情報科学技術を利用した情報ネットワークのおかげで、ネットオークションは、地域的なつながり、職業上のつながりを持たない人同士にも普及した。その結果として、いくつかの「光の部分」そして「陰の部分」も発生している。

ネットオークションでは「アラート」と呼ばれる仕組みも利用されている。これは、新たに出品された物品に、あらかじめ設定された条件に合致するものがあれば、購入希望者にメールを利用して通知する仕組みである。たとえば「A社製カーステレオ用ケーブル」のように、通常は捨てられてしまう商品をアラートとして設定している人がいる。そのことを知る出品者は、一見して価値がない物品でもネットオークションに出品する。それは、誰かのアラートに合致し、売買が成立する。これは、中古品の価値がネットオークションによって変わった一例であるといえる。

ネットオークションの普及により良くなった点をもう一つ述べる。従来は中古商品を売ろうとする専門家が、さまざまな情報を利用して価格を設定していたが、ネットオークションが普及したため、相場（市場価格）がネットオークションに左右されるようになった。結果として、通常の中古市場も含めて、著しく高い額、あるいは安い額での流通が減少し、適正な価格が広く普及するようになった。

ネットオークションの普及は、不法な商品の流通も可能としている。利用者は、たとえば、偽ブランド商品、違法コピーされたソフトウェア、爆発物製造マニュアル、麻薬などの中毒性薬物などの「不法な商品」が出品されることがある。ネットオークション事業者は、不法な商品の出品・売買を禁じていることが多いが、あまりにも多数のオークションが行なわれているため、オークション事業者がすべてを把握できていないこともある。以前は、これらの不法な商品を簡単に購入することはできなかったが、ネットオークションは状況を変えたといえる。

また、以前では想像もできなかった詐欺行為も、ネットオークションが作り出した。ネットオークションにおける取引行為では、通常、決済と商品引き渡しは、オークション業者の手を離れ、出品者と落札者が直接行なう。そのため、送金したのに商品を送ってこない、商品を送ったのに送金されないというトラブルが発生する。オークション事業者もこの種のトラブルを防ぐために、利用者が行なった健全な取引回数を公開し、トラブルが多い事業者が利用しにくい制度を作っている。

しかし、犯罪者はこの仕組みさえも悪用するようになった。たとえば、「チャリンカー詐欺」と呼ばれる手法では、(1) 自作自演で小額商品の売買を繰り返す、(2) 持っていない商品を出品して落札者から入金があったところで、(3) その金額以下で中古品を購入して発送する、などの手法で「健全な取引回数」を増やす。そうして、ある程度の信用を得られる取引回数になったところで、高額商品

を大量にカラ出品し、相応の入金があった時点でメールボックスや口座を閉じ、代金をだましとるのである。

このようにネットオークションには、その光と陰が付きまとうのである。

5.5 情報格差（デジタルデバイド）

本章の最後に情報格差について述べよう。情報格差とは、複数の社会や人の間に存在する、情報環境の格差のことである。格差の要因は複雑であるが、ここでは、主なものを述べる。

1. **経済格差**：いわゆる先進国と呼ばれている国では、コンピュータ・ネットワークを利用することが可能な人が多い。一方、発展途上国では、コンピュータ・ネットワークを購入するよりも、日常の食事や医療費の確保が優先されている人が多い。このような人達は、コンピュータ・ネットワークの恩恵を受けられない。
2. **地域格差・位置格差**：経済格差と無関係な地域格差も存在する。特に、ネットワークが敷設されていない地域でインターネットを利用しようとするれば、通信衛星などの非常に高額な通信手段が必要となる。また、その地域にネットワークが届いていても、屋内にある利用場所までの敷設が困難な場合も存在する。そのため、高速なネットワークを利用できない地域・位置が存在する。
3. **情報リテラシー格差**：情報科学技術の恩恵であるコンピュータ・ネットワークを上手に利用するには、利用者自身が情報リテラシーを身に付ける必要がある。一定の情報リテラシーを身に付けた人は、次なる情報リテラシーの学習機会に出会い、さらなる情報活用の恩恵を受けられる。一方、情報リテラシー教育を受けていない人は、コンピュータ・ネットワークを活用できず、結果として、新たな情報リテラシーを身に付ける機会を逸してしまう。

情報科学技術によってもたらされたコンピュータ・ネットワークを活用できる人達は、より豊かに暮らすことができる。その結果、発展途上国で情報科学技術の恩恵を受けられない人達との格差、高速なネットワークを利用できない地域・位置にいる人との格差、情報リテラシーを身に付けていない人との格差は、さらに拡大する。経済格差、地域格差、リテラシー格差は情報格差を広げ、情報格差は、経済格差、地域格差、リテラシー格差を広げる。「情報格差は自ずと拡大する」といえる。

こうした情報格差を解消するためには、情報格差と他の格差が、互いに強い影響を及ぼす関係にあることを、多くの人が必要である。特に、政治家

や企業経営者などの人間の経済活動に影響を及ぼす立場にある人、教師や芸術家などの人間の精神形成に影響を及ぼす立場にある人は、情報格差の問題を強く認識し、それを解消するために努力をすることが必要である。

情報格差を解消する具体的な活動としては、次のものが挙げられる。

- ・知的所有権による請求の留保：

情報環境に遅れている国における特許や著作物の使用に対して、使用料金を請求しない。

- ・情報機器やネットワーク敷設への援助、開発補助：

「100ドルコンピュータ運動」に代表されるコンピュータ本体の配布や、ネットワーク技術者の派遣、あるいは、情報環境に遅れている国のなかで、国有言語を使用する国の文字コード標準化支援など。

- ・情報リテラシー教育活動への援助：

情報環境に遅れている国の初等中等教育の現場教員を集め、情報リテラシー教育の重要性を説明する活動など。

このような活動を経済的に支援することも、情報格差を解消する活動の一つである。

第6章 なぜ情報学リテラシーなのか

科学技術リテラシーという運動全体の目標は、21世紀を豊かに生きるための智を身に付けることである。もちろん、それは情報学リテラシーでも同じである。ただし、当然、それぞれの分野の特徴もある。では、情報学リテラシーを身に付けることが、とくに本報告に述べてきたような情報科学技術の基本を知ることが、我々にとってどのような意味を持つのだろうか。この点を最後に述べ、本報告のまとめとしたい。

6.1 豊かな社会を築くため

6.1.1 何を理解すべきか

我々が豊かな社会を築いていくために、多くの大人が情報を扱う科学技術について、次のことを理解しておくことが非常に重要である。単なる表層的な知識ではなく、なぜそうなのか、という考察や経験をともなった「実感として」の理解である。

実感として理解しておくべきこと

1. 未完了性 : つねに発達段階。未完成品が市場・社会に出回る。
2. 即時性 : 影響がすぐ現れる。非常に速いスピードで広がる。
3. 汎用性 : 異分野への展開が容易。まったく予期しない応用も。

これらは第1章で述べた情報科学技術の特質を現象面から述べたものともいえる。それぞれの項目を少し解説しよう。

第一の項目は技術の未完了さである。情報科学技術の多くはまだまだ発達段階の技術である。しかも、情報科学技術の特徴として、未完了のまま市場や社会に出回り利用されることが少なくないのである。

例13 : P2Pは禁止すべきか

ファイルの共有方式である P2P 方式 (Peer-to-Peer) は、違法コピーのばら撒きに使われる可能性があるため、ネットワーク上での使用を制限もしくは禁止する動きがある。

しかし BitTorrent のように通常のウェブブラウザで利用されている使い方もある。これは P2P 方式が、まだ発展途上にあり、正しい使い方しかできないような安全な技術にまで成長していないからである。

通常の科学技術であれば、実験や試行を繰り返した上で製品化されたり、実際に使われたりする。ところがソフトウェアは、とくに特別な装置を使わないでも実行できてしまうので、暫定版でも使うことができてしまうのである。

その結果、我々の実生活の上での使用が試行そのものといえる場合もある。つまり、一般の使用者がモルモットになってしまっているのである。これは、そうしたソフトウェアや技術の提供者が作為的に行っている場合もあるが、そうでない場合もある。完璧と思っていたソフトウェアやシステム、あるいは運用方法が実は不十分で、深刻な問題点があった、などということも少なくない。情報を扱う科学技術には、このような脆弱性が内在しているのである。

第二の項目は影響の伝達速度が非常に速いという点である。良い場合も悪い場合もある。これは情報の伝達が電子的に、しかも自動的に行われているからである。伝達が電子的に行われていても、間に人が介在する場合には、速度は格段と遅くなる。また、何らかの制御がかかる場合もある。けれども、インターネット上の一斉配信のように、自動的に行われ、さらにそれが自動的に伝達された場合、あっという間に世界に広がってしまうこともあり得るのだ。

第三の項目は、情報科学の汎用性、そしてそのことに起因した異分野への展開の容易性である。

本報告でも繰り返し述べてきたようにデジタル化により、文書でも音でも映像でも、皆が同じデジタルデータとして扱えるようになった。そのため、デジタルデータを扱う技術は（原理的には）、文書にも、音にも、映像にも適用できるのである。こうした汎用性から、たとえば（音の）デジタルデータに対して開発された技術が、ほとんど変更なく、映像や文書にも適用できる、ということもあり得る。

そのため、今まで想定していなかった利用法の展開、あるいは新たな応用例が出てくる可能性がある。卑近な例でいえば、携帯電話として開発されたものが、ある時から突然、電子財布として使われるのが主流になるかもしれないのである。

6.1.2 誰が理解すべきか

上記の三項目の理解が、なぜ必要なのか、という点は、誰にとって、によって大きく異なってくる。

上記の三項目の実感としての理解は、行政にたずさわる人や企業人のように、社会や組織を動かす人にとくに重要である。コラムNo.4で「大の大人」と呼ばれている人たちである。どういう点で重要なのかを検証してみよう。

- (1) 変化が早く、また変動の幅も大きな情報科学技術に対し、それを扱う制度

やルールが過剰に厳格だったり、あるいは硬直化した物であったら、実際には、とても従うことはできない。そうすると実情に合わせた違反や無視が日常化し、それが不正や過ちを産み出す原因となる。

(2) 何らかの情報システムを導入する場合、発注側の技術的な知識のレベルが重要なのは当然だが、それを指揮する立場にいる人が情報科学技術の未完性や汎用性を意識しないと、結果として無駄なシステムや冗長なシステムを導入してしまうこともある。

(3) 政治や行政にたずさわる人が情報社会のための政策を決めていく際にも、上記の三項目の理解は重要である。

国（政府）がより良いサービスを提供するためには、個人情報を用いる必要がある。しかも小さな政府・行政を目指すためには、そうした個人情報を公開し、企業等がサービスの提供に加われるようにするのが効率的である。しかし、個人情報の公開を安易に行なうのは危険である。

こうしたトレードオフを考慮し、バランスのよい政策・方針を立案し実行するためには、情報科学技術の特質をよく理解し

ておく必要がある。とくに情報環境が大きく変化していることを前提とした設計が必要不可欠である。情報科学技術だけでなく、情報に関する右のような基本的技能や知識は、政策を立案する人には必須だろう。

情報社会で必須の技能や知識

1. ワープロ、メール、ブラウザなどを使う能力
2. 情報社会における倫理やマナー
3. 情報社会における法律：プライバシーの権利や知的財産権など

各個人においても三項目の理解は重要である。まず、様々な情報サービスの重要度と安全度とのバランスを考え、自分に適したものを選択していくために重要である。各種情報サービスは今後はさらに深く生活に密着していくだろう。もちろん便利になる反面、情報漏れの被害などのリスクもある。リスク回避やリスクテイクには、そのリスクが正しく把握されている必要がある。その必要性を感じ、情報を得て、そしてそれを理解するためには、上記三項目の実感としての理解が重要である。

同様なことは自分自身だけではなく家族に関してもいえる。また、親として保護者として、情報社会における子供の躾や教育のためにも、上記三項目の理解は必要だろう。なぜ〇〇がダメなのか、をある程度明確に語れる親であってほしい。

情報科学技術の特質をよく理解しておくことが政策の立案者に必須だと述べたが、情報社会の政策への参画する一般市民にも、これは重要である。政府や行政が進めようとしている情報科学技術の導入に対し、市民として議論に加わっていくためには、そのバックグラウンドとして情報学リテラシーが重要である。

6.1.3 どうやって理解する：情報学リテラシー

先の三項目を実感として理解するには、本報告書で述べたような情報科学技術の原理・仕組み・影響の理解を基とするのがよい、というのが我々の提案である。

極端な場合を考えよう。センスのよい人が情報科学技術の利用で十分な経験を積み、三項目を深く理解できるかもしれない。つまり、情報科学技術の原理や仕組みを学ばなくても、三項目の深い理解は可能である。しかしそれは非常に特殊な場合で、一般の人々に期待するのは不可能だろう。一般には、本報告書で述べたような情報科学技術の原理・仕組み・影響の理解から入るのが最も効率のよい方法と思われる。

「言葉」を知らないとしきしみを持たない、という人の心理がある。植物を見て、その美しさを楽しむには（原理的には）植物の名前を知る必要はないかもしれない。しかし、人は、その名前を知ることとしきしみを得て、植物観察により深い興味を持つようになる。

かなり比喩的な話ではあるが、これは科学技術全般、そして情報科学技術についても同様である。我々はユーザーとして、便利な情報機器、情報システムを使っている。しかし、そうした便利な道具は、その経験に即して、原理・仕組みを学んでいかないと、すぐに（心理的に）ブラックボックス化されてしまう。「わけがわからないけれども便利だから使っている」という姿勢になってしまう。そうすると情報科学技術についての理解がないまま、情報社会の中で流されていくだけになってしまうのである。

これに対する対策として、我々は情報科学技術の原理・仕組み・影響の理解を提案したわけだが、これらも単に天下りの的に教えるのでは、その効果は低いだろう。各個人の生活に関連付け、便利な道具から原理までがつながるような教育や啓蒙活動が必要である。次節で述べるようなプログラミングの体験を奨励することも一つの方法だと思う。

6.2 豊かに生きるため

情報学リテラシーの普及には、情報科学技術の原理・仕組みに、まず、興味を持ってもらうことが重要である。そのためにはプログラミング体験の奨励を一つの方法として提案したい。たとえば、「20行プログラム運動」(皆で20行ほどのプログラムを書いてみましょう!) などというのを展開するのもおもしろいかもしれない。

プログラミングはとても良質な知的楽しみである。一般に自分の創意工夫で何

か作り、そして動かすことの楽しさは大きい。絵画や工作などもその一例であるが、器用さや製作コストなどを考えると、プログラミングははるかに手軽に楽しめる。

プログラミングというと、数式や記号のイメージが強いが、何も従来のプログラムにこだわる必要はない。

何らかの規則で物事を記述すること自体がプログラミングである。という認識から出発してもよいだろう。ただし、単なる規則ではない。状態とか繰り返し、などの概念が入っているのが重要である。

そのためには、一般の人々が試してみやすいプログラミング環境があるとよい。また、いろいろな入り口の試みがあるとよい。ロボットのプログラムやTVゲームの一つとしての販売などもあるだろう（ダウンロードして使う、というだけで抵抗を感じる人も非常に多い）。

ただし、それだけだと「便利な道具」あるいは「おもしろいおもちゃ」の段階でブラックボックス化してしまう可能性もある。たとえば「Squeak eToysがどうやってコンピュータの上で実現できているのだろうか」という疑問は、「きつととっても難しくて複雑なことをしているのかなあ。とても分からないだろう。まあ（知らないでも）いいか」で終わってしまうかもしれない。

こうした点を補足し、原理・仕組みまでつなげる方法もあるとよいだろう。そのためには、プログラミング体験に興味を持ってもらった人へ、もう少し原理的なことを啓蒙・教育する、というフォローアップも大切である。また、次の例のように、逆に非常に低レベルの計算でも何かができる、ということを経験してもらう、という方法もあるかもしれない。

例14：小学生から楽しめるプログラミング

ノート型パソコンの発案などで有名な A.ケイは、初等・中等教育でのプログラミングの体験のためのプログラミング言語 Squeak eToys を開発し、それを用いた教育の試みを行っている。

Squeak eToys は、従来のように文字列でプログラムを書くのではなく、絵（+コメント）を中心にプログラムを組み立てられるようになっており、自分のデザインした絵（アニメ）を動かすプログラムを簡単に作れるようになっている。

Squeak eToys でも日本語でプログラムが作れるが、英語というバリアをなくし、小学生にもプログラミングを経験してもらえるような試みもいくつか行なわれている。

イノベーションのない企業も国もそのうち衰えていく。2000年ごろにイチローが出ていた日産自動車のCMのキーワードは「変わらなくちゃ!」で、それが「変わらなくちゃも変わらなくちゃ!」になったような記憶がある。このようなメタな物云いはテレビのCMとしては異例だったと思う。それはともかく、物質→エネルギー→情報と続いてきた科学技術の主流の流れで、情報技術はまさに「変わらなくちゃも変わらなくちゃ!」というドッグイヤーどころか、マウスイヤーに突入している。

老若男女を問わない一般の人のための情報リテラシーはもちろん重要であるが、このコラムではことさら「大の大人の情報リテラシー」の必要性を強調したい。情報技術イノベーションを發明・実装できるのはフレッシュな頭脳をもつ若者たちである。しかし、それだけでは不十分である。情報技術イノベーションの展開を支えるのが、ここでいう「大の大人の情報リテラシー」だ。

「大の大人の情報リテラシー」とは、日本の社会を実質動かしている人々、つまり、「大の大人」に必要な、これからの日本の情報技術をどうするのかに対する土地勘とか評価能力である。具体的には、(1) 口先だけでなく、情報技術の重要性を認識すること、(2) 現在の情報技術がまだ未熟な技術であること認識すること、(3) 情報技術の変化の速さを知り、畏怖すること、(4) 最初に言ったように、情報技術のイノベーションを、大の大人の責任として先導しなくては企業も国も衰えていくと認識することである。

上記の(2)に関する典型的な説明の一つは、数年前の東大の数学の入試問題「円周率が3.05より大きいことを証明せよ」に見られる数学の筋金入りの簡潔明瞭さに情報科学や情報技術が到底及んでいないことである。情報技術についてなにか説明しようとする、どうしてあんなに長々と説明しないとイケないのか。これは科学技術としての情報学が教育も含めてまだ枯淡の境地に達していないことの証である。

筆者はこれまで何回も「日本のソフトウェア（産業）はどうしてこうもだめなのか」を趣旨とするパネルディスカッション等に駆り出された。日本のソフト売り上げの60%はいまだにカスタムソフトとのことである。1990年に70%と聞いたのとほとんど変化がない。これに比べて欧米のソフトの売り上げの多くはパッケージソフトだという。これは一言で言うと、欧米のソフトのほうで個別のケースから離れて「抽象化」が進んでいるということである。さらに、洩れ聞こえるところによれば、どうもカスタムソフトを発注するお客さんのほうが馬鹿(?)らしい。なので、ソフト産業のほうも手抜きができる(?)。いずれにせよ、こんな話をするパネルやら委員会が連綿と行なわれているところを見ると、日本のソフト（産業）の風土病を嘆く風土病が日本人にはあるらしい。これはつまるところ、「産業界を牛耳る大の大人の情報リテラシー」が欠如しているということなのである。

欠如の事例はほかにもある。日本で情報技術で大成功する会社が出にくいのは、正しい投資が行なわれないからという説がある。これはお金をもっている投資家に「大の大人の情報リテラシー」がないからである。また、日本では情報技術者の地位や社会評価が低いとも言われる。これは、大の大人によって形成される「社会の情報リテラシー」が低レベルだということである。こうして、人材が育たない、集まらないという悪循環が起こる。

ついでなので、さらに口を滑べらせると、最大の「大の大人」であるところの行政の（広い意味での）情報リテラシーが低レベルである。情報に関する行政がいまだにバラバラなのがその現れである。どこの役所も情報が重要と口を揃えているのに、本当の意味での「大の大人の情報リテラシー」が発揮されていない。いまや、戦略的な企業は CIO (Chief Information Officer) をおいて、情報技術時代での生き残りを真剣勝負で賭けている。ほとんどの省庁や自治体に本物のCIOがいなければ、今後もIT投資の無駄遣いが続きそうだ。そのような無駄遣いの後遺症として、情報技術に新しい予算投入が行なわれにくくなっているという症状まで現れ始めている。

これではなんだかお先真っ暗だが、解決法についても触れておかなければなるまい。日本のお得意は黒船来襲であるが、来てからでは多分手遅れである。「多分」とつけたのは、日本には黒船が来ても間に合わせてしまう能力があるみたいだからである。でも、こと情報技術に関しては基礎体力が失せてしまっている可能性がある。

筆者が昔から主張しているのは、どんな形でもいいから、日本発のサクセスストーリーを誕生させることである。こうすれば、人はついてくる。すべての大の大人を突き動かすほどにないにせよ、そういった事例が出てくる見込みはゼロではない。若い人もうまく発掘して育てれば、これからの日本の情報技術を支える存在になってくれるに違いない。ただし、それを可能にするために、「潰すな、育てよ」の長い目をもつ「大の大人の情報リテラシー」のサポートが必要である。

やはり真っ当に情報科学・技術の面白さを万人に理解してもらうことによる底上げが、時間はかかるものの、正攻法だろう。そのためにも、現在の高校の教科「情報」の実施実態は見直す必要があるだろう。私見では、もっと情報科学に力点をおいていいはずだ。また、文部科学省で検討中と聞いているが、早急に小学校、中学校、高校を通じた連続性と一貫性のある情報リテラシー教育の実施に持ち込まなければならない。念のため、これは教育技術のみの問題ではなくて、広い意味での情報学全体が責任をもたなければならない課題である。

もっともこれでは百年の計になってしまいそうだ。本報告でも主張しているが、もっと直接的に、プログラミングの楽しみや喜びを、子供と言わず、手っ取り早く大の大人の近未来予備軍に知ってもらう努力をする必要がある。筆者の20年ほど前のデータによれば、国会議員という真の「大の大人」には囲碁や将棋の有段者が多い。プログラミングを囲碁や将棋のようなメンタルリクリエーションに育てる努力がアカデミアにも必要だろう。

情報学専門部会名簿

笥 捷彦	早稲田大学 理工学術院	教授
渡辺 治	東京工業大学 大学院情報理工学研究科	教授
芦田 昌也	和歌山大学 経済学部	准教授
川合 慧	放送大学	教授
竹内 郁雄	東京大学 大学院情報理工学系研究科	教授
辰己 丈夫	東京農工大学 総合情報メディアセンター	准教授
西崎 真也	東京工業大学 大学院情報理工学研究科	准教授
萩谷 昌己	東京大学 大学院情報理工学系研究科	教授
原田 悦子	法政大学 社会学部	教授
藤田 憲治	日経B P社	編集長
松井 啓之	京都大学 経営管理大学院/大学院経済学研究科	准教授
益子 典文	岐阜大学 総合情報メディアセンター	教授
吉見 俊哉	東京大学 大学院情報学環	教授

「科学技術の智」プロジェクト 研究組織

平成20年3月現在

1. 評議会

有馬朗人（日本科学技術振興財団会長）〔議長〕、赤田英博（日本PTA全国協議会会長）、阿部博之（科学技術振興機構顧問）、石井紫郎（日本学術振興会学術システム研究センター副所長）、井上和子（神田外語大学名誉教授）、金澤一郎（日本学術会議会長・国際医療福祉大学大学院教授）、佐々木正峰（独立行政法人国立科学博物館館長）、鈴木晶子（京都大学大学院教育研究科教授）、遠山敦子（財団法人新国立劇場運営財団理事長）、中村日出夫（全国中学校理科教育研究会会長）、村上陽一郎（国際基督教大学大学院教授）、毛利 衛（日本科学未来館館長）

【以下、企画推進会議委員】

北原和夫（国際基督教大学教養学部理学研究科教授）、伊藤 卓（横浜国立大学名誉教授）、室伏きみ子（お茶の水女子大学大学院人間文化創成科学研究科教授）、長崎栄三（国立教育政策研究所教育課程研究センター総合研究官）、浪川幸彦（名古屋大学大学院多元数理科学研究科教授）、星元紀（放送大学教授）、岩村 秀（日本大学大学院総合科学研究科教授）、笥 捷彦（早稲田大学理工学術院教授）、西田篤弘（元宇宙科学研究所／総合研究大学院大学理事）、長谷川寿一（東京大学大学院総合文化研究科教授）、丹羽富士雄（政策研究大学院大学政策研究科教授）、渡辺政隆（科学技術政策研究所上席研究官）

2. 企画推進会議

北原和夫（国際基督教大学教養学部理学研究科教授）〔委員長〕、伊藤 卓（横浜国立大学名誉教授）〔副委員長〕、室伏きみ子（お茶の水女子大学大学院人間文化創成科学研究科教授）〔副委員長〕、長崎栄三（国立教育政策研究所教育課程研究センター総合研究官）〔事務局長〕、名取一好（国立教育政策研究所教育課程研究センター基礎研究部総括研究官）〔事務局次長〕、天野 徹（科学技術振興機構審議役）、有本建男（科学技術振興機構社会技術研究開発センター・センター長）、岩崎秀樹（広島大学大学院教育学研究科教授）、岩村 秀（日本大学大学院総合科学研究科教授）、小川正賢（神戸大学大学院人間発達環境学研究科教授）、小川義和（国立科学博物館展示・学習部学習課長）、荻野 博（放送大学副学長）、奥林康司（摂南大学経営情報学部教授）、笥 捷彦（早稲田大学理工学術院教授）、川勝 博（名城大学総合数理研究センター長）、熊野善介（静岡大学教育学部教授）、小林 興（帝京平成大学現代ライフ学部教授）、小林傳司（大阪大学コミュニケ

ーションデザインセンター 副センター長大学院教授)、佐々義子(NPO 法人くらしとバイオプラザ 21 主任研究員)、重松敬一(奈良教育大学副学長)、高安礼士(千葉県総合教育センターカリキュラム開発部部長)、高柳雄一(多摩六都科学館館長)、滝川洋二(東京大学教養学部社会連携寄付研究部門客員教授)、永山國昭(自然科学研究機構 岡崎統合バイオサイエンスセンター長)、浪川幸彦(名古屋大学大学院多元数理科学研究科教授)、西田篤弘(元宇宙科学研究所/総合研究大学院大学理事)、丹羽富士雄(政策研究大学院大学政策研究科教授)、長谷川寿一(東京大学大学院総合文化研究科教授)、馬場鍊成(東京理科大学専門職大学院教授)、古田ゆかり(フリーライター・サイエンス リテラシー プロデューサー)、星 元紀(放送大学教授)、堀 裕和(山梨大学大学院医学工学総合研究部教授)、本田孔士(京都大学名誉教授)、美馬のゆり(公立ほこだて未来大学教授)、吉田 浄(日本科学技術振興財団理事)、吉野輝雄(国際基督教大学教養学部理学研究科教授)、渡辺政隆(科学技術政策研究所上席研究官)

3. 専門部会

(1) 数理科学専門部会

浪川幸彦(名古屋大学大学院多元数理科学研究科教授) [部会長]、森田康夫(東北大学大学院理学研究科教授) [副部会長]、新井紀子(国立情報学研究所情報社会関連研究系教授)、石井仁司(早稲田大学教育・総合科学学術院教授)、上野健爾(京都大学大学院理学研究科教授)、岡本和夫(東京大学大学院数理科学研究科教授)、亀井哲治郎(亀書房代表)、國宗 進(静岡大学教育学部教授)、清水美憲(筑波大学大学院人間総合科学研究科准教授)、根上生也(横浜国立大学教育人間科学部教授)、藤木 明(大阪大学大学院理学研究科教授)、真島秀行(お茶の水女子大学大学院人間文化創成科学研究科教授)、三井斌友(名古屋大学名誉教授)、吉村 功(東京理科大学工学部教授)、米田英一(元東芝システムインテグレーション開発部部長)

(2) 生命科学専門部会

星 元紀(放送大学教授) [部会長]、浅野茂隆(早稲田大学理工学術院特任教授) [副部会長]、入來篤史(理化学研究所・脳科学総合研究センターグループディレクター)、唐木英明(東京大学名誉教授)、小林 興(帝京平成大学現代ライフ学部教授)、丹沢哲郎(静岡大学教育学部教授)、千葉和義(お茶の水女子大学大学院人間文化創成科学研究科教授)、本田孔士(京都大学名誉教授)、松本忠夫(放送大学教授)、室伏きみ子(お茶の水女子大学大学院人間文化創成科学研究科教授)、毛利秀雄(東京大学名誉教授)、渡辺政隆(科学技術政策研究所上席研究官)。 [オブザーバー] 加藤和人(京都大学大学院准教授)、長谷川真理子(総合研究大学院大学教授)、和田正三(基礎生物学研究所特任教授)、青野由利(毎日新聞社論説委員 ※平成 20 年 2 月まで)、

(3) 物質科学専門部会

岩村 秀(日本大学大学院総合科学研究科教授) [部会長]、藤原毅夫(東京大学大学総合教育研究センター特任教授) [副部会長]、池本 勲(東京都立大学名誉教授)、伊藤 卓(横浜国立大学名誉教授)、小倉 康(国立教育政策研究所教育課程研究センター基礎研究部総括研究官)、北原和夫(国際基督教大学教養学部理学研究科教授)、小林啓二(城西大学大学院理学研究科教授)、染宮昭義((財)化学技術戦略推進機構常務理事)、辻 篤子(朝日新聞社論説委員)、中山 迅(宮崎大学教育文化学部教授)、花村栄一(千歳科学技術大学光科学部教授)、濱田嘉昭(放送大学教授)、三浦 登(東京大学名誉教授)、横山順一(東京大学大学院理学系研究科教授)、吉野輝雄(国際基督教大学教養学部理学研究科教授)、覧具博義(東京農工大学大学院共生科学技術院教授)

(4) 情報学専門部会

笈 捷彦(早稲田大学理工学術院教授) [部会長]、渡辺 治(東京工業大学大学院情報理工学研究科教授) [副部会長]、芦田昌也(和歌山大学経済学部准教授)、川合 慧(放送大学教授)、竹内郁雄(東京大学大学院情報理工学系研究科教授)、辰己丈夫(東京農工大学総合情報メディアセンター准教授)、西崎真也(東京工業大学大学院情報理工学研究科准教授)、萩谷昌己(東京大学大学院情報理工学系研究科教授)、原田悦子(法政大学社会学部教授)、藤田憲治(日経 BP 社編集長)、松井啓之(京都大学経営管理大学院/大学院経済学研究科准教授)、益子典文(岐阜大学総合情報メディアセンター教授)、吉見俊哉(東京大学大学院情報学環教授)

(5) 宇宙・地球・環境科学専門部会

西田篤弘(元宇宙科学研究所/総合研究大学院大学理事) [部会長]、唐牛 宏(国立天文台光赤外研究部教授) [副部会長]、縣 秀彦(国立天文台天文情報センター准教授)、池内 了(総合研究大学院大学教授)、磯崎哲夫(広島大学大学院教育学研究科准教授)、糸魚川淳二(名古屋大学名誉教授)、大村善治(京都大学生存圏研究所教授)、上出洋介(京都大学生存圏研究所特任教授)、岸 道郎(北海道大学大学院水産科学研究院教授)、斉藤靖二(神奈川県立生命の星・地球博物館館長)、鳥海光弘(東京大学大学院新領域創成科学研究科教授)、廣田 勇(京都大学名誉

教授)、保坂直紀(読売新聞東京本社科学部次長)、水谷 仁(株式会社ニュートンプレス社編集長)、渡部潤一(国立天文台天文情報センター准教授)

(6) 人間科学・社会科学専門部会

長谷川寿一(東京大学大学院総合文化研究科教授)[部会長]、辻 敬一郎(名古屋大学名誉教授)[副部会長]、伊藤たかね(東京大学大学院総合文化研究科教授)、亀田達也(北海道大学大学院文学研究科教授)、木畑洋一(東京大学大学院総合文化研究科教授)、清水和巳(早稲田大学大学院経済学研究科准教授)、隅田 学(愛媛大学教育学部准教授)、利島 保(広島県立広島大学理事)、戸田山和久(名古屋大学大学院情報科学研究科教授)、二宮裕之(埼玉大学教育学部准教授)、長谷川真理子(総合研究大学院大学教授)、早川信夫(日本放送協会解説委員)、廣野喜幸(東京大学大学院総合文化研究科准教授)、間田泰弘(広島国際学院大学工学部教授)、松沢哲郎(京都大学霊長類研究所教授)、松原 宏(東京大学大学院総合文化研究科教授)、松本三和夫(東京大学大学院人文社会系研究科教授)、山岸俊男(北海道大学大学院文学研究科教授)、山本真鳥(法政大学経済学部教授)、渡辺政隆(科学技術政策研究所上席研究官)

(7) 技術専門部会

丹羽富士雄(政策研究大学院大学政策研究科教授)[部会長]、小林信一(筑波大学大学院ビジネス科学研究科教授)[副部会長]、伊藤順司((独)産業技術総合研究所理事/産業技術アーキテクト)、大河内信夫(千葉大学教育学部教授)、佐々木葉(早稲田大学理工学術院教授)、高安礼士(千葉県総合教育センターカリキュラム開発部部長)、田代英俊((財)日本科学技術振興財団/科学技術館企画広報室次長)、中村正和((株)日鉄技術情報センター特別研究員)、名取一好(国立教育政策研究所教育課程研究センター基礎研究部総括研究官)、谷島宣之(日経BP社編集委員)、山崎貞登(上越教育大学学校教育学部教授)、中川尚志(内閣府経済社会総合研究所研究官 ※平成19年3月まで)。「オブザーバー」元村有希子(毎日新聞社科学環境部記者)、

4. 広報部会

渡辺政隆(科学技術政策研究所上席研究官)[部会長]、小川義和(国立科学博物館展示・学習部学習課長)[副部会長]、縣 秀彦(国立天文台天文情報センター准教授)、亀井 修(国立科学博物館展示・学習部学習課ボランティア活動・人材育成推進室長)、木村政司(日本大学芸術学部教授)、野原佳代子(東京工業大学留学生センター准教授)、服田昌之(お茶の水女子大学大学院人間文化創成科学研究科准教授)、横山広美(東京大学理学系研究科准教授)

5. 事務局

長崎栄三(国立教育政策研究所教育課程研究センター総合研究官)[事務局長]、名取一好(国立教育政策研究所教育課程研究センター基礎研究部総括研究官)[事務局次長]

【国立教育政策研究所】

小倉 康(国立教育政策研究所教育課程研究センター基礎研究部総括研究官)、鈴木康志(文部科学省初等中等教育局教科書調査官)、相馬一彦(北海道教育大学教育学部旭川校教授)、人見久城(宇都宮大学教育学部准教授)、阿部好貴(国立教育政策研究所研究協力者)、斉藤萌木(国立教育政策研究所研究協力者)、熊岡昌子(国立教育政策研究所研究補佐員)、国立教育政策研究所総務部

【日本学術会議】

信濃正範(日本学術会議事務局参事官)、廣田英樹(日本学術会議事務局参事官)、成瀬由紀(日本学術会議事務局参事官補佐)、佐野和子(日本学術会議事務局審議専門職)、関 浩子(日本学術会議事務局審議専門職)、生形直樹(日本学術会議事務局審議専門職付)、阿部左織(日本学術会議事務局審議専門職)

【国際基督教大学】

アンドリュー・ドモンドン(国際基督教大学非常勤講師)、原口るみ(国際基督教大学准研究員)、曾根朋子(国際基督教大学物理学教室)

この報告書の利用について

この「報告書」を編集した「科学技術の智プロジェクト」では、「報告書」に書かれていることが、一人でも多くのひとたちにとっての共通の考え方、共通の知恵になっていくことを希望しています。そのために、「報告書」の著作権に関しては、次のとおり取り扱うこととしています。

記

1. 営利を目的としない利用の場合

- ・誰でも、「科学技術の智プロジェクト」のウェブサイトから「報告書」（の一部または全部）をダウンロードして記録媒体に保存し、またはプリントアウトして利用することができます。
- ・誰でも、「報告書」の（一部または全部の）コピー、送信、貸出し、無料配布、もしくは実費での有料配布などの方法による利用ができます。
- ・誰でも、「報告書」（の一部または全部）を変更、改変、加工、切除、部分利用、要約、翻訳、変形、脚色、もしくは翻案などを施して利用することができます。
- ・上記三つの利用をするに際して、「報告書」の著作権管理者の承諾を得る必要はありませんが、出所または出典として「科学技術の智プロジェクト報告書」と記載してください。
- ・上記の利用方法には例外があります。「報告書」には、第三者の著作物を「引用」として使用しています。引用部分については該当箇所にその表示があります。「報告書」としての利用ではなく、この引用部分のみの利用については、上記の利用方法の例外であり著作権法が定める著作権の制限規定にしたがうことになりますのでご注意ください。

2. 営利を目的とする利用の場合

- ・「報告書」の著作権の管理は、「科学技術の智プロジェクト」の代表研究者である北原和夫が行っています。営利を目的として「報告書」を利用される場合には、北原和夫（国際基督教大学教養学部）にまでお問い合わせください。
- ・「引用」その他著作権法が定める著作権の制限規定にしたがって「報告書」を利用されるときには、もとより自由です。

以上

2008年6月

科学技術の智プロジェクト

