

システム開発者からのアルゴリズムへの期待

山本 剛

システム開発を行う企業の立場から見て、アルゴリズムとはどのようなものか。システム開発における一連の流れを踏まえて、開発者、ユーザーの考え方に対してアルゴリズムが与える影響について考察する。また企業としてのシステム開発側からアルゴリズムへの期待と今後のシステムの発展について述べる。合わせてシステム開発や全体の方針に影響があると思われる点を中心として、筆者が関わった開発経験を基に技術を紹介する。

キーワード：システム開発、アルゴリズムの活用、システム技術、システムインテグレーター

1. はじめに

将棋のプロ棋士対ソフトウェアの対決が近年注目されている。将棋ソフトは40年以上前から存在していたが、プロ棋士と勝負できるレベルになったのはここ数年の話である。プロ棋士と将棋ソフトの対局の盛り上がりを支えているのはインターネットの動画中継だ。今年対局では、全5戦の動画中継の観戦者は延べ200万人を超えたという。昨年まではソフトの差し手を代理の人間が盤上で再現していたが、今年からはロボットアームが駒を動かすようになり、人手を借りずに将棋の対局らしくなった。

将棋ソフトの主たる部分は、盤面の状態を節点としたグラフの探索だ。より効率的に、良い解を求めるためさまざまな情報科学の知見が投入され将棋ソフトが作られている。探索を中心に各種の技術を組み合わせる強い将棋ソフトを実現しているが、プロに匹敵する強さだけでは今の対局の盛り上がりはなかっただろう。将棋ファンを中心に多くの人を引きつける魅力は、強いソフトとそのアルゴリズムだけではなく、前述の動画中継やロボットアームなどを含めたプロ棋士と将棋ソフトの対局のための一つのシステムから発せられている。

アルゴリズムとシステムの共生ともいえる関係が成り立っているわけだが、これは将棋やエンターテインメントに限った話ではない。企業で開発するシステムでは、どのような状況なのか。

2. 「あきらめ」を救う

2.1 実現困難という答え

アルゴリズムは人間が現実的な時間で解くことが困難な問題に答えを出してくれるものだ。

筆者はシステムインテグレーターとして、顧客向けのシステム開発を行っている。システム開発の現場では、ある種の「あきらめ」に遭遇することがよくある。「こんなことをシステムが実現してくれたらいいな」、「こんな結果を返してくれれば助かるのに」といった点は本当にたくさんある。だが「システムが肩代わりするのは難しい」、「意に沿う結果を返せるとは限らない」、「実現は困難」、システム開発者から実によく聞く受け答えである。システム開発を行う立場からすると、できないことを引き受けるわけにはいかないのだから、こういった答えになりがちなのだ。このような答えを聞き続けていたら、情報システムに期待を持っているユーザーもそれはできないことなんだ、とあきらめてしまうだろう。当然である。

情報システムが人の生活や経済に大きく寄与していることは言うまでもない。企業であれば給与計算であったり、自社製品の生産管理、個人単位で考えても外出するとなれば地図を検索する、FacebookのようなSNSで遠くの友人の近況を知る、等々だれでも何らかの恩恵を受けている。将棋の対戦相手になってもらう、もあるかもしれない。給与計算は10進数の計算ができれば最低限必要なことはできそうだ。友人の近況も、インターネット上のサーバーに投稿した内容を保存して参照できれば、共有できるだろう。こうした内容をシステム開発して実現できるかと問うと、できますという答えが返ってくる。

システム開発上、これは実現可能な範囲ということ

だ。それでは実現可能と実現不可能の境界線はどこにあるのか。

2.2 アルゴリズムとシステムを隔てる壁

例として出張旅費の精算を行う情報システムを考える。紙で申請していた出張旅費の精算を電子化するシステムであれば、これは通常実現可能と考えられるシステムだ。だが自分でシステムに入力しているだけであれば記入が多少便利になったにすぎない。

だが今日ある多くのシステムのように、経路探索してそのままルートと運賃が申請できるようになれば、これは単に記録ができるようになったということとは全く違う。紙での申請しかなかった時代に持っていけば、驚き、喜んでもらえるに違いない。

経路探索して旅費を自動で入力してくれる精算システムを実現させるには、以下の条件がそろわなくてはならない。

- 旅費の入力、申請や承認を行う出張旅費精算システム
- 経路探索を行うための数理モデルやアルゴリズム
- 出張旅費精算システムの開発者が、経路探索が現実に可能であると知っていること

これらが組み合わせられて初めて、それまでの手書きの帳票管理とは異なる情報システムが世に生まれ出る。経路探索が一般的でない時代に、こんなことができばと思いつきながら、今の技術では実現できない、と言われた、あるいはそう思って「あきらめ」た人はどれくらいいたのだろうか？

経路探索つきの出張旅費精算システムは、まちがいはなく単なる電子化を超えて人を助ける機能を実現している。だが一般的なシステム開発では、なかなかそこまで至らない。

一つには、どこまでが現実的に可能な範囲なのかかわからないといったことがある。理論上計算が可能というだけではシステムに組み込むことはできないし、旅費の1回の計算に2時間かかります、となってしまえば、得られる結果が良くても実際にシステムとして利用する人はいない。そしてシステム開発者は、顧客に対して提案をする段階では、どんなことをどれくらいできるのかという点の見極めに奔走し、効果が期待できるものならばなんとかして欠点になりうる部分をカバーして一つのシステムとして提供できないものかと頭をひねることになる。

次に、効果が測りづらいという点がある。配送計画などで考えれば最適解が常に出せるとは限らないし、ユーザーからの評価では、ドライバーからするとスト

レスがたまるような移動の組み合わせばかりとなれば計画の作成方法に難ありということにもなりかねない。アルゴリズムの特性や人間が使うという点で避けられない問題だが、一定規模のシステム開発において最終フェイズでは顧客の受け入れテストが設けられており、ここで受け入れ不可となるのは開発における一大事であり、何としても避けるべき事態である。

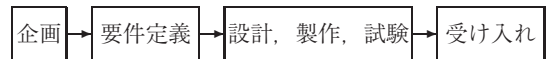
最後に先ほどの条件にも挙げたことだが、そもそも何ができるかわからない、ということがある。残念ながらシステム開発者が研究の現場から情報を仕入れるといったことはそうそうない。この文章をご覧いただいている方に伺ってみたい点だが、このような状況は意外だろうか、それとも常日頃から実感されている状況だろうか？ プログラミングは共通の話であろうと思われるかもしれないが、システム開発のための開発技術とアルゴリズムそのものの実装技術は重なる部分もあるが重ならない部分も多い。

これまで我々システム開発の人間は、日々生まれる新しい知見、手法、アルゴリズムに目を向けていただろうか。研究開発の立場の人間に委ねてよしとしてはいなかったか。今、何かできるのか、できるようになりつつあるのかということを常に追いかけることによって、システム開発との連携の着想が得られるようになる。

2.3 アルゴリズムとシステムの連携

何をしたいか、何を行うべきなのか、これが見えれば、あとは実現のための手段を詰めていけば良い。

システム開発は、以下の段階¹を踏む。



各段階における実施についての課題に対して以下のような解決方法が考えられる。

実現範囲の見極め（要件定義）

要件定義の段階では、性能や何が結果として得られるかを決定する。この決定にあたり、本開発に先駆けて検証を行うためのプロトタイプ開発や性能測定などが非常に有効である。このような準備は開発案件の開始後に着手することは難しい場合も多いため、いつどのように行うかの計画が重要である（企業内の部門にもよるが取り組みが難しい点の一つである）。全般に言えることだが性能などの検証範囲をできるだけ絞り込むための常日頃からの情報収集も欠かせない。

¹ 詳細なプロセスについては、独立行政法人情報処理推進機構による共通フレーム 2013 [1]などを参照されたい。

システム側の工夫（設計、製作、試験）

アルゴリズムがなんらかの点で要件を満たさず、そのままでは使用できない場合の工夫である。設計以降で具体化する。計算に時間がかかるような場合に可能な部分は事前に計算を行って結果を保存しておいたり、いったん受付したことのみに表示してユーザーにメッセージを返すなど、課題・解決法ともさまざまだ。システムとしてはハードウェアを変更、増強することもあるし、速度向上のためにデータベースをインメモリ型にするなどアーキテクチャーでの対応もある。

結果の評価（試験、受け入れ）

難しい点だが、顧客に受け入れられる評価方法を提示することが第一だ。例えば典型的な問題をいくつか用意してその問題を解いた結果をもって判断したり、解に対してユーザーが編集可能なインターフェイスを用意し、すべて人手で行った場合からの短縮時間で評価を行うなどの方法もある。現実の話としては手法に実績があれば納得されるという面もあり、結果の評価は人を強く意識させられる点である。

壁を乗り越えて、システム開発にアルゴリズムの力を借りることで「あきらめ」を救うことができる。これはシステムを支えるアルゴリズムの形である。

3. あきらめの次の希望

アルゴリズムに支えられたシステムは何をもたらすだろうか？ 欲が出るのである。悪い意味ではない。ユーザー側からすると、今まで苦勞していた何かが解決すると思えば、まず使ってみようとする。

3.1 ユーザーの期待

すべては使ってみようというところから始まる。ユーザーの使おうという意志はシステム開発側の想定を超えて広がりを見せる。

使われること

まず使われることが重要である。使われなければ何も始まらないし、開発を行った意味もない。有効に使われなかったり完成まで至らない情報システムも世の中には無数にある。昨今の Web サービスなどではよく浸透している考え方だが、まず使われるということが重要である。システム開発に着手しても本当に使われるという保証はどこにもないし、開発する仕様の検討も最低ラインはその点なのだ。

活用すること

次に単に使うだけではなく活用しようとする人たちが出てくる。当初想定していなかったデータを入力してもらうことで、より計算が速いアルゴリズムの使用を提案したり、システムで得られる情報で今までの業務側のプロセスを改善したりといった利用方法を考案いただくこともある。システム開発側とユーザー側で協力関係が築けた段階と言っても良い。

展開すること

活用に続く広がりもある。同じような課題を抱えている部門や同種の業務への適用など、ユーザー側から見える点を活かして検討いただける。この段階はユーザー側が積極的に利用を促進している段階である。

ユーザーに使おうという思いを呼び起こさせ、正のスパイラルに入ることで、情報システムの導入メリットは何倍にも拡大することができる。そしてこれはユーザー側だけのメリットではない。

3.2 システム開発側の期待

システム開発において、ユーザー・顧客の協力体制を得ることは、開発成功の必須条件と言える（言い換えると協力を得られない開発プロジェクトは失敗する）。協力体制の確立は特に開発初期において重要な課題として力を注ぐ部分であるが、機能に対して前向きな評価をいただいたうえで取り組めるとなれば、非常に大きなメリットである。

またユーザーの協力を得て業務での利用を念頭に置いて、アルゴリズムやシステムの開発内容を選択、改善することができれば、当初の想定に対しての裏付けとより良いシステムを手に入れることができる。そしてユーザーの業務自体の改善に寄与できるのであれば、情報システムが単なる道具の範囲を越えて受け入れられた段階である。そして理想的な状態はシステムの導入が、業務の改善にとどまらず業務の改革あるいはユーザーのビジネス自体を支える存在になることだ。

情報システムには費用対効果の観点が必須であり、各段階でチェックを受けることになる。プランニングの時点で業務の改善といった点まで踏み込んだ成果が見込める場合には計画自体の成立も容易となる。

そして、他部門への適用や同種の業務での利用ということになれば、システム利用のメリットを享受できるユーザーが増えることになり、システム開発側としても新しいビジネスにつながるになれば、これ以上ない成果と言える。

企業にとって、導入部門が増えビジネスの規模が大きくなることのメリットは非常に大きい。またもう1点期待される点としては、受注獲得時の切り札になることだ。情報システムの開発のノウハウ、開発技術などは、基礎的な部分では差がなくなりつつあり、サーバーなどのプラットフォームも汎用品が占める割合が増え、ネットワークも仮想化といった手法が主流になろうとしている。このような状況でシステム開発のみで受注獲得・商談段階で差別化を図ることは非常に困難となりつつある。こういった場合に、他社では実現できないポイントがあることは大きな強みとなる。

単純なシステム開発のみであれば、競争激化、低価格化の方向性は避けられず、付加価値を生み出せる技術やアルゴリズムのニーズはさらに高まっていくだろう。

システムインテグレーターの必要性など昨今問われる部分であるが、ビジネスの観点を除いても技術のコモディティ化や同じようなシステムの焼き直しなど、確かにシステム開発に対して閉塞感を感じることもある。これを打破する一つの方法としても、アルゴリズムとの連携に期待する。

システム開発側がアルゴリズムとその課題解決力に期待する点をまとめると以下となる。

- 課題を解決できることを示し、ユーザーの実現感や前向きな協力体制を引き出すこと
- 業務プロセスの改善、改革や開発計画まで含めた波及効果
- 応用や類似の課題解決への適用による、ビジネス規模の拡大

高いハードルのようにも見えるが、決して夢物語ではなく、ユーザーが大きな期待を寄せるアルゴリズムは確かに存在する。このようなビジネスに寄与する成果があれば、新しい研究成果やアルゴリズムの必要性を訴えることもできる。ここから研究委託等の形で研究の発展に寄与できれば、ユーザー、研究者、システム開発者が幸せになれる循環が生まれる。ユーザーがあきらめを取り去ることで、全員が希望を持つことができる。

月並みではあるが、研究、実用化、ビジネス化のサイクルが回るようにと取り組んでいるが、まだまだこれからの取り組みである。ぜひアイデア、知見、その他お聞かせいただきたいところだ。

4. アルゴリズムの活用を支えるシステム技術

システム開発側からアルゴリズムへの期待を述べた

が、システム開発技術はアルゴリズムにどのような影響を与えられるだろうか？ また我々の持つ開発手法は何か寄与するであろうか？

システム開発側の期待の中で、開発側の技術が平準化されつつある状況について述べた。しかし、そのような状況の中でもシステム技術の使い方次第で工夫が可能な部分や、共通化された技術であっても研究サイドの方には気づきにくい点などが存在する。

アルゴリズムへの期待はすでに述べたとおりだが、システム開発者は何もできないというのでは少々不甲斐ない。システム開発者からの提言として、システム技術が果たすべき役割と、有用な技術やその使い方・場面をご紹介します。

4.1 システム技術の役割

システム技術がカバーするのは冒頭のコンピューター将棋の例で言えば、アルゴリズムによって得られた次の一手を盤上に人間にわかるように指すのはロボットアームだ。情報システムはこのロボットアームの役割を担う。これは単純にユーザーインターフェイスを指すようにも見えるが、それ以外の要素も含む、より広い役割だ。

4.1.1 ユーザーとの対話

ユーザーインターフェイスを含めてユーザーとの接点となる部分である。入力を受け付けて、計算の条件やパラメーターに変換して計算を実行し、結果を人間にわかるように表示する。画面上の対話操作がイメージしやすいが、結果のプリントアウトや緊急性の高い事項はサイレンを鳴らして回転灯を点灯させるような動作まで含む。

実はユーザーとの対話に限れば、必ずしもシステムがアルゴリズムとの橋渡しを行わなくとも良い。この場合の典型的な例はコンサルタントがヒアリングを行い、手法を選択し問題の解を解釈して顧客に伝える方法だ。だがこの方法は社員10万人の企業内の全ユーザーに対しては行えないし、経営層からの依頼だとしても100社の問題を同時には解決できない。だが情報システムになっていけば、全社員向けに利用可能な環境を作れるし、製品として販売すれば100社に成果を届けることが可能だ。コンサルタントに比べれば稚拙ながら、システムがユーザーとの対話を行ってくれているからに他ならない。

また近年重視される点としては、顧客の体験を作り出すためのユーザーエクスペリエンスがある。これはユーザーに機能を使用させるだけではなく、そのシステムを心地よく使うという体験をもたらすためのコン

セプトである。アルゴリズムが結果を計算するならば、システムはユーザーに結果をもらえて良かったという体験が得られる伝え方をすることで、両輪でユーザーの期待を引き出すべきである。

4.1.2 形にすること

効果が見えにくいのだが実は重要な点だ。表側の面と裏側の面の二面がある。

表側の典型的な例として、近年急速に浸透したスマートフォンがある。スマートフォンはアプリケーションをアプリストアから購入して、自分のスマートフォンの一つの機能として取り込むことができる。

ストアで購入して、自分の手のひらに持てるということには実は大きな意味があって、実体のないソフトウェアを自分で所有する感覚が持てるのである。Web インターフェイスに比べると汎用性を捨て去ってしまっているのだが、引き換えに所有感が得られるのである。スマートフォンアプリの形にすると、ある機能を使えることをスマートフォンアプリとして実感を持つことにつながり、結果として利用の促進につながる。単にいつでも使える PC ではないと考え、昨今スマートフォン向けの開発を重視している理由である。

人が使うものとして、ソフトウェアであっても身体性は無視できない。最近の画面の作りとして物理的な動きを取り込んだ表現を加えたり、視線に応じた操作を実現した製品を使われている方も多だろう。これらは単に操作性を向上させるということ以上に、人に対して働きかける力を持つと考えている。

一方裏側の面では、システムが形を得て動作し続けるための下支えとして、データや機能を保全することも重要な役割である。一般に非機能要件と言われる部分で、データストアやバックアップ、リカバリ、可用性向上のための多重化、ネットワーク設計など、システム全体が動作するために必要な事項だ。いずれかの考慮を怠ると、実際に運用していく段階で何らかの問題が発生する。運用時点で発生する直接的な問題はシステムが利用できないということだが、より恐ろしいのはユーザーに使い物にならないと判断されてしまうことだ。これは言うまでもなくユーザーの前向きな姿勢に対して悪影響があるため、運用面では確実な動作のための設計が必要である。

4.2 システム技術の活用

期待される役割に対して、活用できる技術はどのようなものだろうか。

アルゴリズムをシステムの実装が助けることができる部分について、ユーザーとの対話のための技術と、確

実な動作を支えるための技術とを挙げる。広く利用されるという点において、このような実装技術は重要である。アルゴリズムを有効に利活用していただく一助として、これまでの経験を踏まえて手法をご紹介します。

4.2.1 対話を支える技術

対話の中でユーザーインターフェイスは、非常にわかりやすい点である。我々システム開発者がユーザーインターフェイスというときには、画面や入力装置などを指すことが多い。開発のフェーズとしては、同時にデータ構造や方式などの設計も行う。外部からのデータ、例えば小売店の販売予測計算時に参照する天候の情報のようなデータの連携なども含む。画面、データストア、処理方式といった点を組み合わせも含めてユーザーインターフェイスと並行して設計を行う。

操作系で言えば、ユーザーとの対話操作で考えると Web ベースのユーザーインターフェイスが多い。ユーザーインターフェイスというと本質からそれた話と考えられがちだが、例えばアルゴリズムのビジネス化の典型例とも言える Google の検索サービスでも、検索サービスのトップページ上での表示や結果を素早く戻すための工夫は数多く行われている。例を挙げると検索キーワードの入力中に結果やプレビューを先取りしたり、通信プロトコルに踏み込んだ改善まで行っている。Web 上のオンラインの工夫のため、常に取り入れられるというものではないが、彼らがトップクラスのアルゴリズムに胡坐をかいてはいけなく考えているということが重要だ。

システムの実行環境は、ノート PC 1 台上で動作可能なものからスーパーコンピューターが必要なものまでさまざまである。システムを使える形で届けることを考えた場合に、アプリケーションの形態ではインストール作業が必要なことと、インストールそのものに申請、許可が必要であったりするため、可能であれば Web アプリケーション、Web サービス、クラウドサービスの形で提供したい。先に述べたスマートフォンアプリも有力な選択肢だ。

このような提供形態は開発言語の選択に大きな影響があるが、ユーザーの受け入れ度合いで実行環境を増やしたいようなケースでは開発初期の選択が難しい場合もある。この場合は画面周りは HTML、JavaScript ベースで実装する手法が有力だ。計算部分を API の形で分離しておけば、現在ではデスクトップアプリケーション、Web アプリケーション、スマートフォンアプリケーションをほぼ共通のコードで開発可能である。

例えば研究用のアプリケーションをタブレットアプ

リケーションとして開発した後、Web アプリケーションに作り替えることも可能である。フロントエンド側のコードを Web アプリケーションとスマートフォンアプリケーションで完全に共通のものにすることができる。またこの手法であれば前述のユーザーエクスペリエンスについても、昨今の Web サービスのノウハウを取り込みやすいといったメリットがある。

大量データを扱う際など性能面に注意する必要があるが、実装も比較的容易で、実行プラットフォームの追加が想定される開発や研究向けのプロトタイプ開発においても有効な技術である。

4.2.2 動作を支える技術

動作を担保すること、データを保全することなどはシステム開発技術の役割だろう。その他、ネットワーク構成などを含めて運用上の問題が出ないように要件に合わせて設計を行う。通常システム開発では動作し続けることを第一にシステム構成を考えるが、アルゴリズムの利用をにらんでシステム構成の設計を行う場合もある。

例えば、夜間バッチ処理用のサーバーを確保したうえで余剰となる時間帯はこのサーバーを計算用に使うといった構成などが考えられる。大規模な計算環境であるようなタイムシェアリングを一つのシステム内で行うようなものだ。資源が潤沢に使えれば問題ないのだが、実際には限られた予算や資源内で効率良く計算を行うための工夫は必須といえる。

またバックアップは慎重に設計しておかないと、なかなかうまく運用されない部分だが、リカバリとバックアップ、さらにインストールを不要にする手法も存在する。

サーバーの仮想化が進み、仮想マシンのイメージを配布するバーチャルアプライアンスの形態が生まれた。バーチャルアプライアンスは仮想マシンの実行環境があれば OS のイメージを配置するだけでセットアップ済みの仮想マシン内のアプリケーションを利用できるもので、インストール、バックアップ、リカバリを不要とする。システムやアプリケーション自体の配布を行う場合の形態として、非常に有効である。

4.2.3 アンチパターン

失敗から学べることもある。一見、大きなメリットがないように見える実装方法でも、何らかの理由があって採用されていることがある。失敗を避けるための手法は、過去の事例から積み重ねられていることが多い。

ここでは簡単な例を挙げる。

以前にあったケースでは、最初は研究段階で使用するためファイルベースで計算条件などを保存していた。計算条件が管理可能な量である間は良かったが、過去の条件での計算や任意の条件に設定してのシミュレーションの実行など、システム側の機能の拡充に伴い指定された計算条件を呼び出すことも非常に困難となった。ファイルではなくデータベースに計算条件を格納していれば特に問題はなかったのだが、必要な計算条件を探し当てる機能の拡張が困難な状況に陥っていた。この例はシステム開発では定石とも言えるデータベース、データストアを採用したほうが良いケースであった。

問題点をすべて予見することは難しいが、システム開発においてこれまでに蓄積されたノウハウやセオリーは有用なものも多い。特に安定動作に関する手法は数多く、うまく利用すると良いだろう。

5. おわりに

システム開発は、システム開発者と利用するユーザーがそれぞれの立場で関係する。アルゴリズムが開発者とユーザーに対して与える影響と、システム開発におけるアルゴリズムの活用について述べた。またシステム開発の進み方を踏まえて、開発側からの視点で実装時に考慮する点や、有用な手法などを紹介した。

実装技術はアルゴリズム自体の実装とは直接関係がない部分が多いかもしれない。しかし従来のシステム開発の範疇では、課題解決やビジネス面での広がりなど、行き詰まりを感じる点が多々出てきている。アルゴリズムとシステム全体が連携し、相互に恩恵を受けられる形の発展を願う。本稿中、どこか一部でもこんな考え方もあるのだといった気づきにつながれば幸いである。

システム開発とアルゴリズムの組み合わせは、テーマとして取り組んでいるが道半ばである。ユーザーからこんなものかと思われるようなシステムでは、開発していて面白くない。面白くないと言っても各方面からお叱りを受けそうだが、開発者にもユーザーにも驚きと感動を与えるシステムの実現は本当に難しい。この難しさを飛び越える力をアルゴリズムに期待している。

参考文献

- [1] 共通フレーム 2013, <https://www.ipa.go.jp/sec/publish/tn12-006.html> (2014 年 6 月 27 日閲覧)