

序論

TRIZ の歴史

TRIZ (「発明問題解決の理論」) は、ロシアの特許科学者で偉大な発明家である Genrich S. Altshuller (ゲンリッヒ S. アルトシュラー) によって独自に開発された。アルトシュラーは、何十万もの特許を調べた結果、大部分の問題が、様々な時期に様々な文脈で繰り返し解決されてきた [本質的に同じ問題である] ことに気付いた。そして、解決のパターンを分析し、すべての発明が 40 の発明原理のみを適用して導き出されていることを見いだした。また、技術トレンドが極めて予測可能であることも見いだした。これらの原理、戦略およびトレンドを適用することによって、どんな発明的な問題をも解決できることが分かってきた。彼は一連のツールや技法を発明し、それらが今日 TRIZ (ロシア語で *Teoriya Resheniya Izobretatelskikh Zadach*、英語では Theory of Inventive Problem Solving) として知られている。

TRIZ のアプローチ

TRIZ は、どんな問題をも問題定義から解決策生成に至るやり方で解決する体系的アプローチを含む。その問題解決の方法は ARIZ (「発明問題解決のアルゴリズム」) として明瞭に説明されている。問題解決への TRIZ アプローチはおおよそ以下のステップで構成される。

1. もとの問題を再構成する: 問題を解く第 1 ステップは、問題を定義することである。(問題がこの段階ですでに解決されてしまったことが分かり、さらに先に進む必要がなくなることがよくある。これは、問題とされていたものがもはや問題の様相を呈しなくなってしまうからである)。システムを分析して、機能、リソースおよび対立を見いだす。また、問題を非常に明確な用語で記述する。

本ステップの後半は、究極の理想解(IFR)または究極の理想目標をイメージすることである。(多くの場合、理想的な結果は達成可能でないかもしれないが、究極の理想解の定式化は正しい方向に解決策を考えるのを助ける)。究極の理想解の実現を阻むシステム内の対立または矛盾は何か [を考える]。

2. 矛盾を解決する: 第 2 ステップは、矛盾を解消することである。矛盾を解決するのに矛盾マトリックスと発明原理を利用することができる。このステップは、標準解、(発明)原理、技術進化トレンドなどのさまざまな TRIZ 技法、および特許データベースなどの知識源を使用することによって、代替案を見つける過程を伴う。TRIZ は世の偉大な発明から知識と経験を借用することを教え、また、勧める。

3. 解決策を分析する: 次のステップは、解決策を比較検討し、劣った解決策を捨てて優れた解決策を選択することである。空間、時間および環境条件を勘案して優劣を判断する。優れた解決策は自ら矛盾を解決し、有害な要素を役立つリソースに変えるものである。また、外部リソースを必要とする代わりにシステム内にある既存のリソースを利用する。

TRIZ 技法とその適用

TRIZ はいくつかのツールと技法で構成されている。主な技法は、発明原理、矛盾、理想性、標準解と技術進化トレンドである。これらの 5 つはあらゆる TRIZ 専門家により一般的に受け入れられており、踏襲されている。他にも、リソース、機能分析、"物質-場"分析、システムオペレータ、ARIZ および他の TRIZ のツールや技法など、大いに強力な問題解決ツールであって、TRIZ 実践者の一部に活用されているものがある。各種ツールが問題解決プロセスの様々な段階で使用されており、ツールによっては、特定タイプの問題で、より役立っているものもある。

Umakant Mishra 著、

『IT とソフトウェアにおける問題解決アイデア集 – TRIZ の発明原理で分類整理』 (仮題)
(Mishra 「IT & TRIZ」 翻訳プロジェクト、仮訳原稿、2008 年 1 月 27 日、小西慶久・中川 徹)

技法のいくつかはあまり使用されていなかったり、TRIZ 専門家の中で意見が分かれたりしているが、それでもなお特定の場合には強力でありえる。それぞれの技法の詳細は、本書の範囲を超えているので、立ち入らない。われわれは、発明原理のみに焦点を絞ろう。それは TRIZ の最も基本的な技法とみなされているものである。

TRIZ 技法は当初、機械的問題の分野から導出されて適用されてきたが、その後、他の分野の問題に対しても適用可能であることが分かってきた。そこには、宇宙、エレクトロニクス、化学、財政、マーケティング、人的資源、環境、そして IT (情報通信技術) などの分野を含む。しかし、分野の特性により、さまざまな TRIZ 技法の解釈と適用が多少変わってくる。本書では、目的とする範囲から逸脱しないよう、IT における適用に限って議論しよう。

発明原理

発明原理の適用は、TRIZ の諸技法の中で、最も簡単で、それでいて最も強力な技法の 1 つである。TRIZ の研究結果によると、発明原理は 40 個しか存在せず、それらが今まであらゆる発明で適用されてきた。換言すれば、同じ 40 の発明原理が今後も、あらゆる種類の発明的問題の解決に適用可能ということになる。この事実から、TRIZ は非常に強力な、問題解決に必須の手法として位置づけられる。

各(発明)原理は特有のアプローチで問題に対処する。いくつかの(発明)原理が互いに重なり合っているようにも見えるが、これは限られた場合のみに起こることであり、(発明)原理の有用性が低下するわけではない。ある状況に対して適用できる発明原理が 1 つに限られる必要はなく、むしろ、多くのケースにおいて発明原理の組み合わせや集合が適用可能である。例えば、犬の吠え声 [の録音] (発明原理 26: 「コピー」) は、泥棒に対する警報(発明原理 9: 「先取り反作用」)として、犬から切り離して(発明原理 2: 「分離」)、用いられている。

TRIZ 研究者は発明原理の適用性をさまざまな研究分野にどんどんと拡大してきた。その中には、機械工学、微生物学、化学工学、製薬、企業経営、ソフトウェア工学などがある。その成果を論文や書籍として既に発表している研究者もあるが、さらに多くのものが近い将来発表されるであろう。

IT における発明原理

TRIZ の 40 の(発明)原理が IT の様々な分野で広く使用されていることがこれまでの研究結果から明らかになった。筆者は 1,000 以上の IT 特許を分析し、各発明において 40 の発明原理の適用性を確認した。

本書の文脈において、IT という用語を、ハードウェアまたはソフトウェアのみに限定すべきではない。IT という用語をより広義に捉えて、ハードウェア、ソフトウェア、プログラミング、データ処理、ストレージと管理、ネットワークと電気通信、インターネット技術、プロジェクト管理、プロセスと品質の管理、などを含むものとする。

40 の(発明)原理は基本的には任意のソフトウェア問題に適用可能であるが、既存の文献における説明はソフトウェアの文脈において時として不適當である。例えば、「相変化」、「熱膨張」、「強い酸化剤」などのいくつかの(発明)原理は、ソフトウェアの文脈では非常に紛らわしい。同様に、「パラメータの変更」のような(発明)原理は、ソフトウェアにおいても大変明確に適用できるものの、その説明に用いられている「物理状態を変える」あるいは「温度を変える」などの表現はソフトウェア環境にはそぐわない。これは、ソフトウェアの文脈で使用される「パラメータ」が物理的物体で使用されるパラメータと異なっていることに起因する。本書では、(発明)原理の背後にある基本概念を変えずに、これらの記述のいくつかをソフトウェア環境において意味があるように表現しなおした。

本書では、(発明)原理の一つ一つを順次議論し、さまざまな例とケーススタディを用いて、その(発明)原理がどこでどのように適用されたかを明らかにしよう。なお、解決策の実際の実現方法が時と場合によって異なりえることを、心に止めておくことが大事である。例えば、発明原理 1: 「分割」は、複数の人に仕事負荷を分配できること、ソフトウェアをいくつかのモジュールに分けられること、そして、プロジェクトをいくつかのフェーズに分けられることを示唆してくれる。しかし、実際にソフ

Umakant Mishra 著、

『IT とソフトウェアにおける問題解決アイデア集 – TRIZ の発明原理で分類整理』 (仮題)
(Mishra 「IT & TRIZ」 翻訳プロジェクト、仮訳原稿、2008 年 1 月 27 日、小西慶久・中川 徹)

トウェアをいくつかのモジュールに分けるか、そして、各モジュールに何人を割り当てるかを決めるのは、究極のところ自分に任されている。もちろん、(発明)原理の適用をさらに進めて厳密解への到達を試みることも可能であるが。

発明原理に関するいくつかの事実

40 の発明原理は互いに排他的ではない。状況によっては 2 つ以上の発明原理を適用しているようにも見える。このような場合、どの発明原理を適用するか迷ったり、(発明)原理の正確な定義を議論したりするのはではなく、問題の解決に専念すべきである。発明原理の適用は(他のあらゆる TRIZ 技法の適用と同様に)、問題を解決するという目的に対するひとつの手段にすぎないのだから。

多くの場合、同じ(発明)原理に対して別名がある。例えば、「抽出原理」は「分離原理」とも呼ばれ、また、「反転原理」は「逆発想原理」とも呼ばれたりする。これらの「正式名称」の違いは、ロシア語から英語への翻訳上の問題であり、それ以外の解釈をするべきではない。

あらゆる(発明)原理が同じ頻度で使用されているわけではない。分割、分離、汎用性、先取り作用、ダイナミック性、フィードバック、仲介、コピー、パラメータ変更など、他の(発明)原理よりも頻繁に使われているものもある。

発明原理は、あらゆる研究分野で適用することが基本的には可能である。しかし、これらは元々、機械的あるいは技術的な問題に対して開発されたものだから、その説明に用いている言葉が他の分野の状況には関連していないように見えることもある。このような状況においては、発明原理の背後にある概念や論理を変えることなく、その分野において適用できるように表現し直す必要がある。

本書の意図

TRIZ に関しては多くの良書があるが、それらは TRIZ の概念と技法に関する説明を試みており、応用指向のものはほとんどない。さらに、IT の分野においては、TRIZ および発明原理の IT 分野との関連性および適用性について記述しているまともな研究資料がほとんど無い。本書がこのような不備を補い、先行者として将来の研究者や発明家を導くことができれば幸いである。

発明原理について説明している数多くの文献があるが、それらは皆、工学と技術の言葉を用いて(発明)原理を説明している。本書の目的の一つは、発明原理についての既存の説明を見直し、ゲンリッヒ・アルトシュラーによって開発されたオリジナルの 40 の(発明)原理の概念、構造、およびアプローチを変えることなく、IT の文脈に沿ってそれらを修正することである。

本書は、各発明原理に対する多数の適用例と発明から構成されており、すべての IT 発明家、問題解決者および解決策開発者にとって必須の参考文献となることを意図している。また、本書が、「ソフトウェア(発明)原理」あるいは「IT(発明)原理」の貴重な参照源となることを意図している。

本書の構成

本書は、非常に平易で口語的な言葉を用いて、経験豊富な IT 専門家でも初心ファンでも、どのようなレベルの読者にも簡単に楽しめるようにしている。本書には日々の IT 操作に関わる何千もの例が含まれている。随分単純で自明に見えるものもあろうが、適用可能領域が分かるように、それらも不可欠のものとして含めてある。TRIZ を批判する人は、このような周知の例や技法を集めてどこが良いかと、非難することができよう。まったくもってその通りである。われわれはみんな日々の活動において、知らず知らずのうちに発明原理を適用してしまっているのだから。

発明原理はわれわれの知識や実践とさほどかけ離れてはいない。実際、発明原理は、われわれの知識や実践から一般化されたり抽象化されたりして、導出されてきたものである。新たなことを教えるというよりも、既に存在するものとその可能性について思い起こさせるという考えに基づいている。

本書では、(米国特許データベースから選択した) 100 個の IT 発明と IT 産業からの 100 のケーススタディを、各発明原理に均等に分配して示している。特許を引用する目的は、(発明)原理が適用可能であ

Umakant Mishra 著、

『IT とソフトウェアにおける問題解決アイデア集 – TRIZ の発明原理で分類整理』 (仮題)
(Mishra 「IT & TRIZ」 翻訳プロジェクト、仮訳原稿、2008 年 1 月 27 日、小西慶久・中川 徹)

ることを明確に示すためである。特許はすべて米国特許データベースから取った。さまざまな分野や概念を網羅するべく、特定の特許権者や企業を重視せずは無作為に選出した。

本書を章順に読む必要はない。各章は独立しているので、好きな順に読むことができる。巻末の索引は網羅的であり、(発明) 原理をさまざまな IT 解決策に関連づけるのに参照できる。

本書ははっきりと発明原理にだけ焦点を合わせており、TRIZ の他の技法 (例えば、矛盾、理想性あるいは (進化)トレンドなど) は含んでいない。これらの他の技法に弱点があるとか、無関係だとかということではなく、これらは今後の研究の題材とし、別の著書としたいと思っている。

本書の対象読者

本書は、IT 分野における発明や問題解決に興味を持っているすべての人を対象としている。だれが読んでも理解できるように言葉遣いをできるだけ簡単にしている。本書は 40 の章で構成され、各(発明)原理に対して 1 つの章を割り当てている。各(発明)原理は何百もの例といくつかのケーススタディを用いて説明している。読者が IT 分野にどの程度関わっているかにより、それぞれの例は簡単に映ったり難しく思えたりするであろう。

世界中の何千人もの TRIZ 研究者、TRIZ 学生と指導者、あらゆるレベルの IT 専門家、解決策開発者、意思決定者、アナリストや建築家が、各自の現実問題を解決するのに本書が大いに役立つと感じて気に入ってくれることを願っている。また、本書が IT 発明家や IT 特許の書き手にとって不可欠な伴侶となれば幸いである。

この本がひとりでのわれわれの問題を解決できはしないことを申し上げておきたい。問題に気づき、定義するのは人間だから、問題を解決するべきものも人間である。本書は、問題を解決するための技法がどのようなものであるか、そして、類似の問題が業界においてどのように解決されたかを示している。発明原理により、問題のさまざまな側面から見ることができ、問題に対するさまざまな解決策を知ることができる。したがって、本書は、いかなる IT 問題をも解決する知識と能力をわれわれに究極的に与えてくれる。

Umakant Mishra

2006 年 8 月