

# コンピュータ分野における TRIZ 矛盾表の適用

## アーキテクチャ・方式など向きの矛盾表・発明原理

庄内 亨 (日立製作所 中央研究所)、河辺 峻 (明星大学\*)、

濱中 直樹 (日立製作所 マイクロデバイス事業部\*)

### 概要

TRIZの矛盾マトリクスと発明原理をコンピュータ特許創生に適用した際の問題点とその対策について述べる。現状で提示される発明原理の詳細は機械、構造、物理、化学といった分野に偏っており、この発明原理から連想してコンピュータ特許のアイデアを出すには、具体例を積み重ねることにより、連想しやすい言葉を増やしていくことが重要である。また、39x39の技術矛盾マトリクスは大きく扱いにくい、関係の薄いパラメータを思い切って削除することによりコンピュータのアーキテクチャ・方式に特化したサブマトリクスを作成でき、有効に使用できる。適用事例として、インターネット・データセンタに関する問題解決についても述べる。

### 1. はじめに

発明的に問題を解決するための理論としてTRIZ が普及しつつある。しかし、このTRIZ は「コンピュータやソフトウェア分野に使えるのだろうか?」という疑問の声が良く聞かれる。我々のうちの一人が1999年にTRIZの調査を開始したときにも同様の疑問を感じた。以来、我々はコンピュータやソフトウェア分野への組織的適用のための努力を重ねて来た。2001年後半に発明原理をソフトウェア分野の用語で読み替える試み[1, 2]もあったが、分野が限定されており、また、我々が対象とする分野とも異なり、直接の活用には困難があった。

本論文では我々の活動の中から、技術矛盾マトリクス(矛盾表)と発明原理をコンピュータ分野の特許創生に適用した際の問題点とその対策について報告する。なお、「コンピュータ分野」と言ってもその範囲は一般には広く、我々が扱ったのは、主に、プロセッサや並列計算機のアーキテクチャ・方式・論理、回路方式、ストレージ装置(RAID、NAS)、データベース管理システム、ITシステム管理、画像・音声・言語処理などである。

技術矛盾マトリクスと発明原理を眺めると分かるが、そこに出現する言葉は機械、構造、物理、化学といった分野に偏っており、コンピュータやソフトウェア分野へ適用しようとするとき困難を感じる。具体例を積み重ねることにより、連想しやすい言葉を増やしていくことが重要である。まず、技術矛盾マトリクスの改良/悪化パラメータがコンピュータ分野のどんな用語と対応するかを示す必要がある。また、技術矛盾マトリクス(旧版)は39x39と大きく扱いにくい

いが、関係の薄いパラメータを思い切って削除することにより、コンピュータのアーキテクチャ・方式などに特化したサブマトリクスを作成でき、有効に使用できる。更に、現状で提示される発明原理の詳細も分野が偏っており、具体例を積み重ねることにより、発明原理から連想してコンピュータ特許のアイデアを出すための連想しやすい言葉を増やすことができる。

これらを実施した後、適用分野をストレージ装置、ITシステム管理、画像・音声・言語処理などに拡大して、技術矛盾マトリクスの修正と発明原理の具体例の拡充等を行った。

適用事例として、筆者らが関わりあっていたインターネット・データセンタ(IDC)に関する問題解決についても述べる。このアイデアの一部は日立のサービスプラットフォームコンセプトHarmonious Computingや統合サービスプラットフォーム BladeSymphonyに活かされたものである。

本論文と似た分野の仕事としては、ソフトウェア工学とTRIZの関係を論じた研究[3-5]、ITプラットフォーム分野でのTRIZ適用[6]やソフトウェア工学向けのTRIZ[7]などもある。[3-5, 7]はソフトウェア開発プロセスを主に扱っており、我々の分野とは異なり、直接の活用には困難がありそうである。[6]は組織的取組みを中心とした発表であり、適用事例の収集・周知活動の重要性を指摘しているが、事例の紹介はあまりない。また、[8]では、SカーブによるITシステムの進化の分析例や、コンピュータではあまり進歩しない要素技術(メモリアクセス時間など)を上位のシステム技術(アーキテクチャ技術)で解決している(キャッシュの例など)など、コンピュータ分野でのシステム分析

\* 本研究は日立製作所中央研究所に在籍していた時のものである。

アプローチの有効性の事例を提示している。

なお、最新の矛盾マトリクス Matrix 2003[9]では改良/悪化パラメータや矛盾表の頻度データがコンピュータ分野をも考慮したものになっているが、コンピュータ分野にほとんど関係しないパラメータは依然含まれている。また、発明原理は以前のままであり、本研究のアプローチは Matrix 2003に対しても有効と思われる。

## 2. 言葉の変換、サブマトリクス化

TRIZの概要を知ってコンピュータ特許創生に適用しようとすると、analysisとsynthesisの2ステップになる。Analysisでは特許の対象問題・アイデアを、TRIZを用いて解析する。SynthesisではTRIZが提示するヒントに基づき具体的発想をする。直感的にも

- ・ analysis は有効
- ・ synthesis が課題

であることが判る。Synthesis を成功させるためには、

- ①コンピュータ分野向けの言葉に変換
- ②技術的矛盾マトリクスの適用分野へのサブマトリクス化が必要と考える。

そのため、実際の特許創生の場面に参画し、具体的な特許の問題・アイデアにTRIZを適用した。

### 2.1 具体例の積上げ

まず適用可能性検討の初期に実施した実際のコンピュータ分野、特にアーキテクチャ・方式や回路分野の特許事例4件の分析について述べる。この分析では、発明者がTRIZを用いずに創生した特許アイデアをTRIZを用いて分析している。

#### 2.1.1 プロセッサ[10]

【内容】高性能サーバ向けキャッシュアクセス制御機構。マルチバンク構成のキャッシュメモリにおいてバンク競合を回避するために、バンク競合を起こすロード命令の組を予測し、バンク競合を起こさないと予測される組で2つのロード命令を同時発行する。

【TRIZ適用】

- ・悪化パラメータ → 6 面積が増える(マルチポートキャッシュによるチップ面積の増加)
- ・改善パラメータ → 25 時間の損失(性能を上げたい)
- ・提示される発明原理 → 10 アクションの先取り、35 状態の変移、17 他次元への転換、4 非対称性

【特許アイデアの核心】予測テーブルを前回までの履歴に応じて作成しておく。それを見て競合を起こさないロード命令の組を予測し、発行する。

【TRIZ的分析】35 状態の変移 を予測することにより 10 アクションの先取りを行う。

【解釈・コメント】「状態の変移を予測することによりアクションの先取りを行う」という形の発想は分岐予測やメモリアダプター予測などを初めとするコンピュータの高速処理における基本的考えである。他に発明原理として 17 他次元への転換、4 非対称性 が提示されている。これに基づく他のアイデアがあるかも知れぬ。

#### 2.1.2 高速キャッシュアクセス方式[11]

【内容】SMP(Symmetric Multi Processing、対称型マルチプロセッシング)ではプロセッサ数の増加に伴い、Store のスループット低減が大きな課題である。そのためにはStore-in-cache が必須である。実装の制約により、すべてのプロセッサでキャッシュを完全共有するのは困難で複数のStore-in-cache が必要となる。複数のStore-in-cache間の一致制御のオーバヘッドを減らしたい。

【TRIZ適用】

- ・悪化パラメータ → 24 情報が失われる(一致制御の失敗)
- ・改善パラメータ → 25 時間の損失(性能を上げたい)
- ・提示される発明原理 → 24 仲介、26 コピー化、28 方式の転換、32 色を変える

【特許アイデアの核心】データの参照形態を参照履歴を用いて予測し、その結果を活用して、動的にキャッシュ一致制御を最適化する。

【TRIZ的分析】データに 32 色をつけ(データの参照形態を参照履歴を用いて予測し)、そのデータに関しては26コピー化して共有化する。

【解釈・コメント】32 色を変える という発明原理がたびたび登場する。コンピュータのアーキテクチャ・方式・論理にとって、これは「情報にタグをつけて他と区別する」という意味に解釈すべきで、非常に有効な発明原理である。

#### 2.1.3 キャッシュメモリ構成方式[12]

【内容】キャッシュメモリの容量には上限があり、限られた容量の中で、如何にしてヒット率を向上させるかが課題である。局所性の高いデータを見つけ、区別してヒット率を向上させたい。

【TRIZ適用】

- ・悪化パラメータ → 24 情報が失われる(一致制御の失敗)
- ・改善パラメータ → 25 時間の損失(性能を上げたい)
- ・提示される発明原理 → 24 仲介、26 コピー化、28 方式の転換、32 色を変える

【特許アイデアの核心】ある種のデータをその他のデータより優先的に格納でき、どれを優先するかを決めるポリシーをシステムやパーティション毎にプログラムの性質を考慮して変更することのできるキャッシュメモリ構成方式。

【TRIZ的分析】データに 32 色をつけ(局所性の高いデータにタグをつけて他と区別する)、24 仲介ポイント(切り替えポイント:パーティション毎、システム構成時、ブート時、...)毎に変更可能にする。

【解釈・コメント】24仲介 という発明原理もたびたび登場する。コンピュータのアーキテクチャ・方式・論理にとってみると、ブロック転送発生時とか、割り込み発生時とか、レベルによりいろいろ考えられる。何かの切り替えポイントと考え、そのポイントを活用する特許アイデアがある。

#### 2.1.4 位相調整回路[13]

【内容】LSI チップ間の信号インタフェース回路、特に装置間インタフェースのようにデータビット間での遅延時間ばらつきが大きいインタフェース回路において、スキューばらつきを改善したい。

【TRIZ適用】

- ・悪化パラメータ →9 チップ面積が増える
- ・改善パラメータ →31 悪い副作用(スキューばらつきを

改善したい)

- ・提示される発明原理→22 害を益に変換、1 細分化、40 複合材料

【特許アイデアの核心】送信側から伝送されるデータ信号と並送クロック信号との位相差を複数点で検出して保持し、その情報を基に送信側からのデータ信号の位相を調整する。

【TRIZ的分析】22 害(位相差大)をみて益(位相調整)に変換。1 細分化(部分化により詳細に制御)。

【解釈・コメント】この矛盾マトリクスにたどり着くのに少し時間がかかった。そもそも 31 悪い副作用 とは抽象的で良く判らないが、この場合は「スキューばらつきが拡大する」がこれに相当すると考える。提示される発明原理は 22 害を益に変換 であるが、この特許アイデアを見るとピッタリである。しかし、「害を益に変換」という言葉だけでこのアイデアが生まれるためには、相当の技術的専門知識が必要である。

#### 2.2 具体例分析のまとめ

このような実際の特許創成の現場への参画により、発明原理の幾つかを、コンピュータのアーキテクチャ・方式向けの言葉に変換することができた。例えば、「32 色

悪化パラメータ→ 改善パラメータ↓	6	9	10	12	15	16	24	25	26	27	30	31	33	34
	不動物体の面積	速度	力	形状	動く物体の運動の持続性	不動物体の運動の持続性	情報の損失	時間の損失	物質の量	信頼性	物体に働く有害要因	悪い副作用	操作の容易さ	保守の容易さ
6 不動物体の面積	+	-	1,18,35 .36	-	-	2,10,19 .30	30,16	10,35,4 .18	2,18,40 .4	32,35,4 .04	27,2,39 .35	22,1,40	16,4	16
9 速度	-	+	13,28,1 5,19	35,15,1 8,34	3,19,35 .5	-	13,26	-	10,19,2 9,38	11,35,2 7,28	1,28,35 .23	2,24,35 .21	32,28,1 3,12	34,2,28 .27
10 力	1,18,36 .37	13,28,1 5,12	+	10,35,4 0,34	19,2	-	-	10,37,3 6	14,29,1 8,36	3,35,13 .21	1,35,40 .18	13,3,36 .24	1,28,3, 25	15,1,11
12 形状	-	35,15,3 4,18	35,10,3 7,40	+	14,26,9 .25	-	-	14,10,3 4,17	36,22	10,40,1 6	22,1,2, 35	35,1	32,15,2 6	2,13,1
15 動く物体の運動の持続性	-	3,35,5	19,2,16	14,26,2 8,25	+	-	10	20,10,2 8,18	3,35,10 .40	11,2,13	22,15,3 3,28	21,39,1 6,22	12,27	29,10,2 7
16 不動物体の運動の持続性	-	-	-	-	-	+	10	28,20,1 0,16	3,35,31	34,27,6 .40	17,1,40 .33	22	1	1
24 情報の損失	30,16	26,32	-	-	10	10	+	24,26,2 8,32	24,28,3 5	10,28,2 3	22,10,1	10,21,2 2	27,22	-
25 時間の損失	10,35,1 7,4	-	10,37,3 6,5	4,10,34 .17	20,10,2 8,18	28,20,1 0,16	24,26,2 8,32	+	35,38,1 8,16	10,30,4	35,18,3 4	35,22,1 8,39	4,28,10 .34	32,1,10
26 物質の量	2,18,40 .4	35,29,3 4,28	35,14,3	35,14	3,35,10 .40	3,35,31	24,28,3 5	35,38,1 8,16	+	18,3,28 .40	35,33,2 9,31	3,35,40 .39	35,29,2 5,10	2,32,10 .25
27 信頼性	32,35,4 0,4	21,35,1 1,28	8,28,10 .3	35,1,16 .11	2,35,3, 25	34,27,6 .40	10,28	10,30,4	21,28,4 0,3	+	27,35,2 .40	35,2,40 .26	27,17,4 0	1,11
30 物体に働く有害要因	27,2,39 .35	21,22,3 5,28	13,35,3 9,18	22,1,3, 35	22,15,3 3,28	17,1,40 .33	22,10,2	35,18,3 4	35,33,2 9,31	27,24,2 .40	+	-	2,25,28 .39	35,10,2
31 悪い副作用	22,1,40	35,28,3 .23	35,28,1 .40	35,1	15,22,3 3,31	21,39,1 6,22	10,21,2 9	1,22	3,24,39 .1	24,2,40 .39	-	+	-	-
33 操作の容易さ	18,16,1 5,39	18,13,3 4	28,13,3 5	15,34,2 9,28	29,3,8, 25	1,16,25	4,10,27 .22	4,28,10 .34	12,35	17,27,8 .40	2,25,28 .39	-	+	12,26,1 .32
34 保守の容易さ	16,25	34,9	1,11,10	1,13,2, 4	11,29,2 8,27	1	-	32,1,10 .25	2,28,10 .25	11,10,1 .16	35,10,2 .16	-	1,12,26 .15	+

図 1 矛盾サブマトリクス

を変える」という発明原理がたびたび登場する。コンピュータのアーキテクチャ・方式にとって、これは「情報にタグをつけて他と区別する」という意味に解釈すべきで、非常に有効な発明原理である。また、「24 仲介」という発明原理もたびたび登場する。コンピュータのアーキテクチャ・方式にとってみると、ブロック転送発生時とか、割り込み発生時とか、レベルによりいろいろ考えられる。何かの切り替えポイントと考え、そのポイントを活用する特許アイデアが有り得る。

一方、改良/悪化パラメータの言い換えでは、「性能を上げたい」は「25 時間の損失」、「論理規模の増大、チップ面積の増大」は「6 静止物体の面積」、「キャッシュ貫性の保障」は「24 情報が失われる」に対応させられることがわかった。

また、コンピュータのアーキテクチャ・方式にとって関係の薄いパラメータを思い切って削除し、必要と思われる14のパラメータのみを抽出し、14x14の技術的矛盾サブマトリクスを作成した。サブマトリクスを作成することによりマトリクスの面積が1/4以下にすることができ、A4一枚やPCの800x600の画面に収めることができ、初心者でもTRIZの適用がやり易くなる。

### 3. 適用範囲の拡大

上記の特許事例はコンピュータ(プロセッサ)のアーキテクチャ・方式や回路方式の分野に限定されていたが、その後、分野を広げて、特許創成の現場で実際にTRIZを適用した問題解決・発想を行った。対象分野は、並列計算機の方式、ストレージ装置(RAID、NAS)の方式、データベース管理システム、ITシステム管理、画像・音声・言語処理などである。この過程で、改善・悪化パラメータや発明原理の言い換え、発明原理への事例の追加を行った。また、サブマトリクスに関しては、分野を拡大すると2.2で述べた改善・悪化パラメータでは不足する事例が出現したので、32 製造の容易さ、36 装置の複雑さ、37 検出と測定の困難さ、39 生産性 などのパラメータ追加を行った。

発明原理の言い換えは各分野の技術的知見をエッセンス化するのに相当する。ある程度の期間に渡って特定の技術分野で研究・開発を続けた研究者には、その分野、その部門での技術的問題解決の際に頻りに活用される解決戦略が暗黙知として蓄積されている。発明原理の分野ごとの言い換えは、この暗黙知を明文化することに相当する。これを用いることにより、その分野に精通していない人に、その分野の主要な解決戦略を伝授できる。

## 4. コンピュータ . . . . .

最後に、コンピュータ特許創生への適用の試みの事例として、筆者らが関わりあっていたインターネット・データセンタ(IDC)に関する問題解決について述べる。1999 年頃、特に米国において高速インターネットに接続されたデータセンタ(IDC)の構造が注目を浴び始めていた。当時は、インターネットでのアクセス集中による応答性能の悪化等も問題として顕在化し始めており、2000年には国内でもi-modeがシステム障害となってサービス停止にも発展し、社会的にも大きな問題として取り上げられていた。これらを始めとするデータセンタの課題に対応するため、米国における状況および社内の先端ユーザへのヒヤリングを通じてIDC の問題点と解決方式の検討に入った。

### 4.1 IDC . . . . .

IDC の解決すべき問題の一つは次のように捉えることができる。

- ① インターネットからの急激な負荷に対応してサーバ構成を変化させ、最適化によりスループットを上げる。
- ② サーバ数(全体の物量)を減らす。(コスト削減のため)
- ③ 最適化を操作性良く行う。

解決すべきIDC のシステム構成の一例を図2に示す。

多くの解決方式を具体的に検討する中から本命となる解決方式を考案した。「仮想プライベート・データセンタ」というコンセプトである。仮想プライベート・データセンタはVirtual Private Data Center (VPDC)の訳でこれ以降はVPDCと略す。VPDC はデータセンタ内にある機器(サーバやストレージなど)を複数のユーザや業務で共有して利用効率を向上させると共に、予想を越えるインターネットからのアクセス集中への対応を可能にするデータセンタ・アーキテクチャである[14, 15]。このコンセプトの基本特許をとろうということになり、TRIZを適用した。

### 4.2 . . . . . (TRIZ . . .)

解決すべき問題点①②③を一気のTRIZの矛盾マトリクスに適用するには無理がある。そこでまず①②を合わせて考える。この2つは、

■ サーバ構成を変化・最適化してスループットを上げたい。かつサーバ数(物量)も減らしたい。  
という矛盾を解決する問題である。

【TRIZによる分析】

悪化パラメータ →26 物質の量(サーバ数が増える)  
改善パラメータ →25 時間の損失(スループットを上げたい)

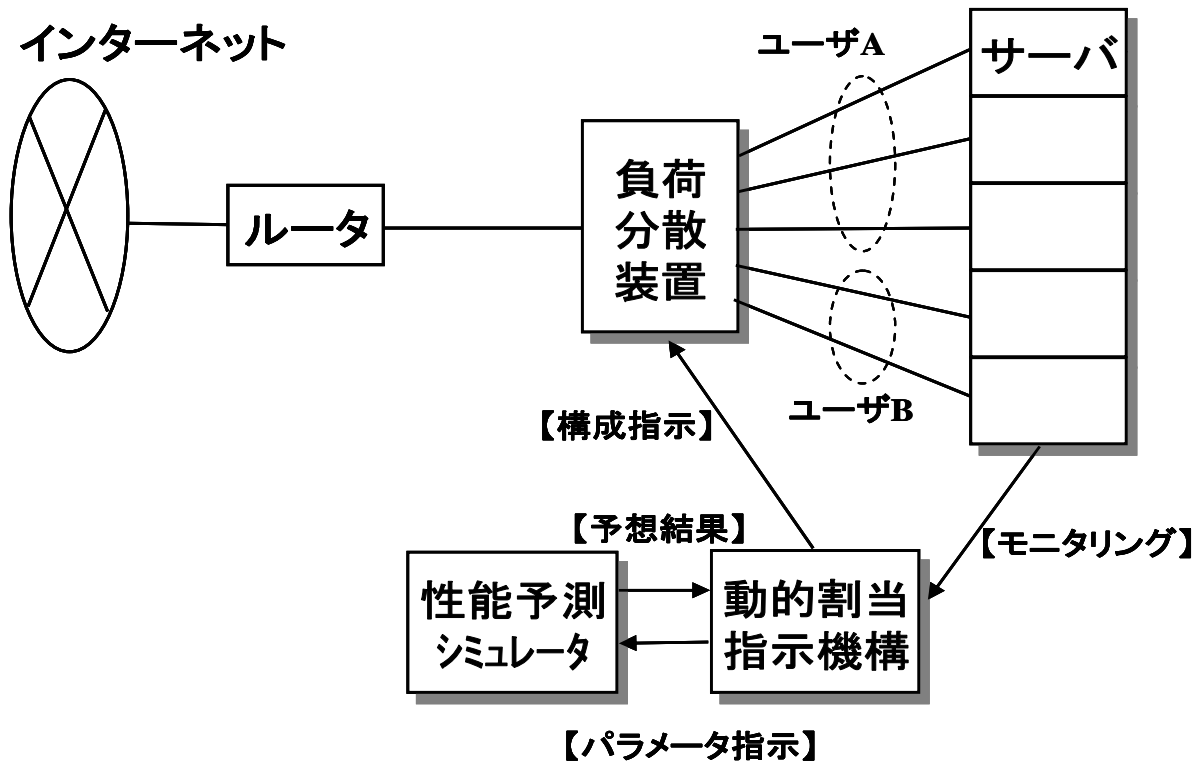


図2. 解決すべきiDCのシステム構成

提示される発明原理→35 状態の変移、38 強力酸化剤を使う、18 振動・発信・高周波、16 部分的解決または過剰解決

提示された発明原理のうち、38は全く関係がないと考えられるからTRIZ的に考えると、『状態の変移をとらえて(予測して)部分的解決を図るか、または過剰解決を図る、その際に周期的・波動的な動きを考慮する』という発明のヒントが得られる。考えられる状態の変移は、サーバのCPU効率の変移、サーバの主記憶効率の変移、サーバのI/O 効率の変移などであり、これらがモニタリングされることが望ましいが、少なくともCPU効率の変移は絶対必要である。また、部分的解決または過剰解決に関しては限られたリソース(サーバ数)の中での最適配置を考える問題となる。これは解決すべき問題点③と合わせて、TRIZが利用できる。

■ 最適配置を簡単に行う。⇒操作性良くしたい。(改善パラメータ)

かつ、操作性良くしようとすると時間がかかる。(悪化パラメータ)

【TRIZによる分析】

矛盾マトリクスと提示される発明原理

悪化パラメータ →25 時間の損失(最適計算する時間がかかる)

改善パラメータ →33 操作の容易さ(最適配置を簡単に

行いたい)

提示される発明原理→4 非対称性、28 方式の転換、10 アクションの先取り、34 部分の放棄・変形または再生成。

提示された発明原理からTRIZ的に考えると、モデル化によるシミュレーションや実測を行い、先を予測することにより、先行して最適配置をする(アクションの先取り)こと、サーバ等の放棄・再利用により最適配置をする(部分の放棄・変形または再生成)ことが、発明原理としては有効であることを示している。この結果、TRIZより得られる問題解決のヒントは、

- ・状態の変移をとらえて(予測して)部分的解決を図るか、または過剰解決を図る、
- ・モデル化によるシミュレーションや実測を行い、先を予測することにより、先行して最適配置をする(アクションの先取り)、
- ・サーバ等の放棄・再利用により最適配置をする(部分の放棄・変形または再生成)、
- ・周期的・波動的な動きを考慮する、など

となる。これを発明のヒントとしてアイデアを出すことになる。実際、TRIZを適用したことにより、より大きな完成したアイデアが生まれた[16, 17]。矛盾マトリクスから必要な発明原理が提示され、それを元にいろいろ考えることにより、アイデアのバリエーションが充実化され、より完成したア

アイデアに近づけることができた。

このようにTRIZは発明のヒントを与えてくれると共に、考える的を絞ってくれることが重要なポイントである。TRIZから得られるものはここまでで、後はこれらのヒントから発明が創生されるかどうかである。この時点でアイデアが出るかどうかは、考えている人達の専門知識の深さに大きく依存する。アイデアが出ない場合、それはTRIZに問題があるのではなく、考えている人達に問題があるとも考えられる。

なお、ここでのアイデアの一部は日立のサービスプラットフォームコンセプトHarmonious Computingや統合サービスプラットフォーム BladeSymphonyに活かされたものである。

## 5. おわりに

矛盾マトリクスや発明原理をコンピュータ分野の特許創生へ適用した際に必要であった発明原理の言い換えやサブマトリクス化について述べた。これらは、TRIZを組織的に適用する際に有効である。しかし、これらだけで十分な効果が得られる訳でもない。技術課題やアイデアの表層のみの理解に基づくTRIZの適用では効果はなく、技術や問題の全体を深く把握・理解し、状況に応じて最適なTRIZツールをうまく適用し、従来の思考方法も捨てずに活用することも重要である。

なお、最新の矛盾マトリクス Matrix 2003[9]では改良/悪化パラメータがコンピュータ分野をも考慮したものになっているが、コンピュータ分野にほとんど関係しないパラメータが依然含まれている。また、発明原理は以前のままであり、本研究のアプローチはMatrix 2003に対しても有効と思われる。

・・・ 本稿に対して有益な助言を下された重田淳二氏に感謝いたします。

.....

[1] K. C. Rea, TRIZ and Software – 40 Principle Analogies, Part 1, The TRIZ Journal. Sept, 2001, <http://www.triz-journal.com/archives/2001/09/e/index.htm>.

[2] K. C. Rea, TRIZ and Software – 40 Principle Analogies, Part 2, The TRIZ Journal. Sept, 2001, <http://www.triz-journal.com/archives/2001/11/e/>.

[3] 中川 徹、「ソフトウェア工学とTRIZ (1) 構造化プログラミングをTRIZの観点から見直す」, 『TRIZホームページ』, 2004年 8月, <http://www.osaka-gu.ac.jp/php/nakagawa/TRIZ/jpapers/>

2004Papers/NakaSE1-StrProg040824/NakaSE1-StrProg040824.html

[4] 中川 徹、「ソフトウェア工学とTRIZ (2) 段階的詳細化をTRIZの観点から見直す」, 『TRIZホームページ』, 2005年 2月, <http://www.osaka-gu.ac.jp/php/nakagawa/TRIZ/jpapers/2005Papers/NakaSE2-StepRefine050207/NakaSE2-StepRefine050207.html>

[5] 中川 徹、「ソフトウェア工学とTRIZ (3) ジャクソン法をTRIZの観点から見直す」, 『TRIZホームページ』, 2005年 2月, <http://www.osaka-gu.ac.jp/php/nakagawa/TRIZ/jpapers/2005Papers/NakaSE3-Jackson0509/NakaSE3-Jackson050906.html>

[6] 鈴木、木村、神山、江本、高井、渋澤、NECにおけるTRIZ推進活動、第2回TRIZシンポジウム、2006.

[7] D. Mann, Re-Structuring TRIZ To Meet The Needs Of Software Engineering, 第2回TRIZシンポジウム、2006.

[8] 河辺峻、インターネット時代のコンピュータアーキテクチャ入門、第6章、近代科学社、2004.

[9] D. Mann, S. Dewulf, B. Zlotin, and A. Zusman, Matrix 2003 Updating the TRIZ Contradiction Matrix, CREAM Press.

[10] 中村、島田、プロセッサシステム、特許3956652、特開2002-366350.

[11] 細谷、山本、キャッシュアクセス制御方法およびデータ処理システム、特開2002-32265.

[12] 川本、樋口、キャッシュ装置、特開2002-140234.

[13] 齊藤、インターフェース回路、特開2002-9748.

[14] M. Masuda, T. Tarui, T. Shonai, S. Kawabe, M. Sugie, "VPDC: Virtual Private Data Center – A Flexible and Rapid Workload Management System," The Eighth IFIP/IEEE International Symposium on Integrated Network Management (IM 2003) pp. 473-476.

[15] 吉村、垂井、庄内、河辺、「Webアクセス集中対応データセンタプラットフォームの研究」、信学技報、FTS 2001-29, pp. 33-40.

[16] 玉置、庄内、佐川、河辺、計算機資源分割装置および資源分割方法、特開2002-24192.

[17] 吉村、垂井、マシエル、庄内、計算機資源割当方法、特開2003-124976.