

解決策の生成法 (3) 解決策を生成する方法の体系 (USIT)

目標: TRIZをやさしく・使いやすくすることを目指したUSIT法における解決策の生成法を学ぶ。
特に、USITの「解決策生成技法の体系」を学び、その考え方と使い方を例を通して学ぶ。
単なる「発想法」でないアイデア生成の考え方の本質を理解する。

前回: 解決策の生成法 (2) 「壁」を破る方法 (ブレイクスルー)

目標: 問題解決の核心は、「壁」(困難・矛盾)を打ち破って、「ブレイクスルー」を達成することである。そのための具体的な考え方を、TRIZおよびASITから学ぶ。

要点: TRIZでは、「壁」のことを「矛盾」と呼び、敢えて「矛盾」を明確化する。
最も突き詰めた段階が「物理的矛盾」(システムの一つの面に対して、正・逆の対立する要求が同時にある)であり、この「にっちもさっちも行かない」状況を
TRIZは「分離原理」で確実に解決できるといい、多くの適用例がある。

また、アルトシュラーの「小さな賢人たちによるモデリング」の思考法も有用。

ASITはTRIZをずっと簡略にしたもので、5種の解決策生成法だけを使う。

「創造的解決策であるための2つの条件」=

閉世界制約条件 (問題中に違う種類のオブジェクトを持ち込まない)

質的变化の条件 (問題となる効果と(一つの)属性の関係が質的に変化する)

また、演習として、「授業をよりよくする方法」について継続して検討した。

なお、前回資料の「超節水型水洗トイレ」のLeeらによる解決策を以下に資料として示す。

アルトシュラーの方法 ('分離原理'による'物理的矛盾'の解決)

要求を、はっきり言え。

S字管が、「在る」ことを要求する。
S字管が、「無い」ことを要求する。

これは矛盾だ。
にっちもさっちも行かない。

(1) これらの要求を、時間、空間、その他の条件で分離できないか?

時間で分離できる。

「在る」要求は、通常時いつも。

「無い」要求は、水を流すときだけ。

(2) 分離した時間帯で、各要求を完全に満たす解決策を作れ。

通常時間帯: S字管を存在させる。

水を流す時間帯: S字管は存在させない。

(3) そして、両者の解決策を組み合わせさせて使え。

さて、どうしたら
いいのだろう?

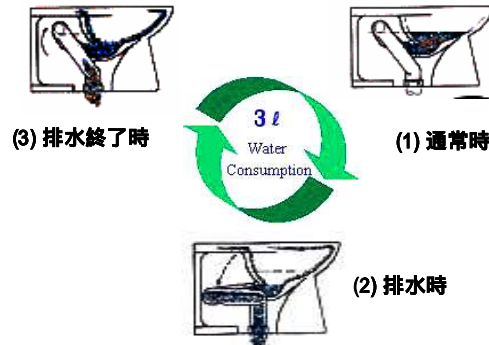
通常はS字管が存在し、水を流すときはS字管が存在しない。

考え方のヒント: S字管の存在/消滅

→ S字の状態か/そうでないか

→ 途中が高くなっているか/なっていないか

解決策: 固定的なS字管をやめて、
プラスチックの管をつけて、水を流すときに下げる。



効果: 消費水量は約3リットル 「超節水トイレ」

2. USITの解決策生成技法の概要

USIT法 (Unified Structured Inventive Thinking) の歴史的経過:

1993年 フォード社の Ed Sickafus が イスラエルの SIT 法と TRIZ に接する。

1995年 Sickafus が SIT 法を改良して USIT を作り、社内教育を開始。
教科書を出版 (1997年)

1999年 中川が Sickafus のセミナーで学び、日本に導入。以後日本で改良・発展。

- 文献: 中川 徹: 「やさしいUSIT法を使ってTRIZのエッセンスを教え・適用した経験」, TRIZCON2002 論文 (2002年4月); 『TRIZホームページ』掲載
中川 徹, 古謝秀明, 三原祐治: 「TRIZの解決策生成諸技法を整理してUSITの5解法に単純化する」, ETRIA 国際会議論文 (2002年11月); 『TRIZホームページ』, 2002年9月。
中川 徹: 「USIT 2日間実践トレーニングセミナーのやり方 (やさしいTRIZの普及のために)」, 『TRIZホームページ』, 2005年7月。
中川 徹: 連載(全5回)「USIT入門: 創造的な問題解決のやさしい方法」, 『機械設計』誌、2007年8月~12月号; 『TRIZホームページ』掲載。
中川 徹: 「日本におけるUSITの発展 - 創造的な問題解決の新しいパラダイム - 」, 第4回日本TRIZシンポジウム、2008年9月、『TRIZホームページ』掲載。

USIT法には、つぎの5種の解決策生成技法がある。(中川, 2002)

- (1) オブジェクト複数化法: オブジェクトを「複数化」する。
英語での「複数」の概念は、(1以外) 0, 2, 3, ... ∞ , 1/2, 1/3, ... 1/ ∞ を含む。
オブジェクトをなしにする、2個、3個、... にする、
1/2 ずつにして一緒に使う、新しいものを導入する、... など
- (2) 属性次元法: オブジェクトの属性の「次元」を変える。

有害な性質を使わなくする、有用な性質を新たに使う、
空間によって性質を変える、時間によって性質を変える、... など

- (3) 機能配置法: 機能をオブジェクト間で「再配置」する。
あるオブジェクトの機能を別のものにさせる、
あるオブジェクトの(複合した)機能を分割して他に担わせる、
複数の機能を統合して一つのオブジェクトに担わせる、
新しい機能を導入する、... など
- (4) 解決策組み合わせ法: 二つの解決策をいろいろな面から「組み合わせる」。
空間的に組み合わせる、時間的に組み合わせる、
機能的に組み合わせる、... など
- (5) 解決策一般化法: 各解決策の概念を「一般化」し、階層的に体系化する。

これらの解決策生成法は、「何を」「どうする」の形で述べている。

それぞれは「オペレータ (演算子)」である。

このオペレータを、対象となるすべての要素に順次適用して、その結果を考える。

これらを繰り返し適用すると、多数の解決策のアイデアが得られる。

最初の3種は、システムの3基本概念 (オブジェクト, 属性, 機能) に対応している。
後の2種は、得られた「解決策」に対してさらに適用するものである。

このUSITの「解決策組み合わせ法」は、TRIZの「分離原理」によく対応する。

「分離原理」= 「矛盾する要求を分離し、別々に満足させて、組み合わせる。」

「組み合わせ法」= 「複数の解決策を (さまざまに) 組み合わせる。」

「組み合わせ法」の方が (「分離原理」よりも) 考えやすい。

組み合わせのいろいろなやり方を説明しやすい。考え出しやすい。

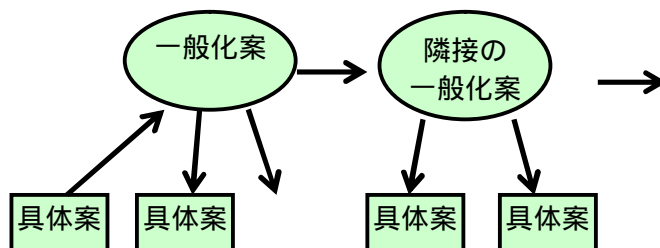
「矛盾の認識」と「分離」のプロセスにこだわらないで先に進むことができる。

「解決策一般化法」は、解決策中の用語・概念を一般化させるものである。

特に、特定の用語、技術用語を、より広い、より総括的な用語で置き替える。

この用語の置き替えによって、概念が膨らむ (より広がる)。連想ができる。

何らかの解決策の具体案が浮かぶと、つぎの図式でアイデアを膨らましていく。



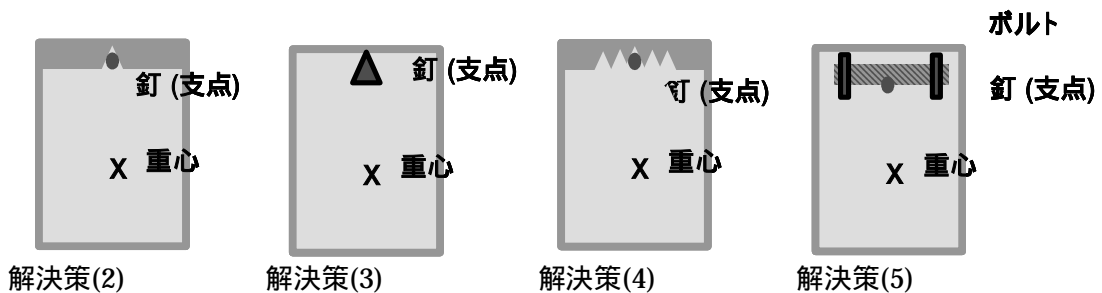
3. USITの解決策生成法の簡単な適用例

「額縁掛けの問題」: 額縁が傾かない (傾きにくい) ような「額縁掛け (の方法)」を作れ。
 (Sickafus (1997) & 中川 徹 (2001)) 『TRIZホームページ』, 2001年

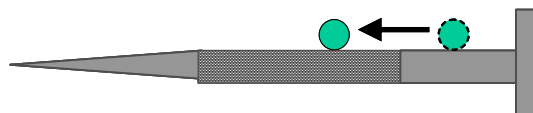
3.1 いろいろな解決策のアイデアとUSITの解決策生成法の理解

この問題の Sickafus の教科書の記述を, 中川が整理して分かりやすくしたもの:

- (1) まず単純化してみる。重心のずれがない完全な額縁を考える。(=< 属性次元法)
- (2) 重心のずれがなければ, ひもの調節が不要だから, ひもを無くす。
 額縁の上枠の真ん中で釘にちゃんと掛ければよい。(=< オブジェクト複数化法)



- (3) 重心のずれがあるときには, 釘に掛ける位置を少し調節してずらせばよい。
 (=< 属性次元法) [釘を掛ける (左右の) 位置という属性]
- (4) 額縁の上枠に釘を掛ける凹みを多数作っておけばよい。(=< オブジェクト複数化法)
- (5) 額縁の上枠の釘を掛ける凹みを, ボルトを使って回せば連続的に調整できる。
 (=< 属性次元法) [凹みの位置属性の値を時間変化]
- (6) 壁からの振動を小さくするために, クッション材を入れる。(=< 属性次元法)
 具体的には, 額縁と壁の間にスポンジ, ゴムなどを入れる。
- (7) 額縁と壁の間の摩擦を大きくし, 動きにくくする。(=< 属性次元法, 機能配置法)
 具体的には, 粘着材, 両面テープなど。
- (8) ひもが釘のところで滑らないように, 釘の面を粗くする。(=< 属性次元法)
- (9) 釘の表面に, 粗い部分と滑らかな部分を作り,
 ひもを調節するときには, 滑らかな部分を使い,
 調節が終わると, ひもを釘の表面の粗い所に動かして固定する。



演習: この解決策はUSITの解決策生成法のどれを使ったのだろうか?

回答 (中川 徹, 2001年11月): つぎの5種の方法のどれとでも解釈できる。

- (a) オブジェクト複数化法: 「釘」オブジェクトを半分ずつにして, 性質を変えて統合。
- (b) 属性次元法: 釘の表面の「滑らかさ」の属性の値を, 釘の部分によって変えた。
- (c) 機能配置法: 釘の「調節」機能と「保持」機能を分離し, 釘の別部分に担当させた。
- (d) 解決策組合せ法: 釘を滑らかにして調節しやすくする解決策と,
釘の表面を粗くして, 傾きにくくさせる解決策とを,
釘の部分分割することにより組み合わせた。
- (e) 解決策組合せ法 (その 2): 上記二つの解決策を時間によって組み合わせた。

「一つの解決策を複数の解決策生成法の観点で同時に解釈できる」のは大事な認識。

これは, システムの 3 つの基本概念「オブジェクト, 属性, 機能」が,
それぞれ単独では存在しないことによる。

だから, 解決策生成法の説明でも, 観点の主たる対象への操作を述べ,
その細目 (具体的なやり方の指針) で他の基本概念に言及している。

よって, USIT の 3 種の解決策生成法は, 実際には重複しあっている。
考えやすい解決策生成法を使えばよい。

3 種の方法を繰り返し使うので, より確実に解決策のアイデアが得られる。

3.2 USIT の「解決策一般化法」を用いた連想的思考の例

例: 「額縁掛けの問題」において, つぎのような考えの連鎖が得られた。

- (壁からの振動があると, 額縁が傾きやすい)
- > 額縁と壁の間にスポンジのようなものを入れよう。
 - > ゴムでもよいね。柔らかい方がよい。
 - ==> 壁からの振動を小さくするために, 額縁と壁の間にクッションを入れる。
 - > クッションは額縁に貼りつけてもよいし, 壁に貼りつけてもよい。
 - ==> 壁を柔らかくする。
 - ==> 壁が柔らかくて, 額縁が滑りにくいともよいかも。
 - ==> 壁と額縁の間の摩擦を大きくする。
 - > 額縁の下辺の後ろに, 粘着剤を塗る。
 - > 額縁の下辺と壁とを, 両面テープで軽く止める。
 - ==> 壁の広い面が, くっつきやすければ, 額縁をペタッとくっつけばよい。
 - ==> 「くつつく壁」のイメージ
 - > マジックテープのようなもので, 額縁を壁にくっつける。
 - > 磁石を額縁に仕込んで, 壁にくっつけてもよい。
 -

このような「連想」的な発想のアイデアを大事にして, それを記述していく。

記述の道具はいろいろ:

黒板, ホワイトボード, 電子記録式のホワイトボード,
模造紙, 模造紙とポストイットカード, ノート, ...

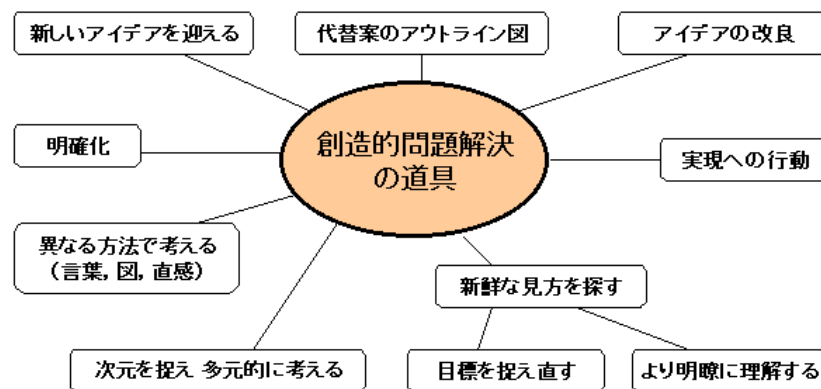
途中で追加, 挿入, 書き換え, レイアウト変更, 図式化が容易なものが望ましい。

この「解決策一般化法」の長所:

連想を刺激し、一般化と具体化との組み合わせで、アイデアが急速に膨らむ。強力。
アイデアを広げることを助ける。
得られる解決策が、自然に体系化・階層化されて整理される。
アイデアの抜け・落ちを見つけやすい。(網羅的にしやすい)
U S I T法の Particle 法における「行動-性質ダイアグラム」と親近性がある。

なお、この一般化と連想の図式は、これだけでも (U S I Tとは別に) 強力な方法である。
欧米では最近、「マインドマッピング」と呼ばれることがある。

テーマを中心に描き、そのまわりに、同心円の放射状にアイデアを書いていく。
思いつくままに記述し、図の配置を修正しつつ、拡大させていく。



例: 出典: 中川 徹: 『T R I Zホームページ』, 1999年2月。

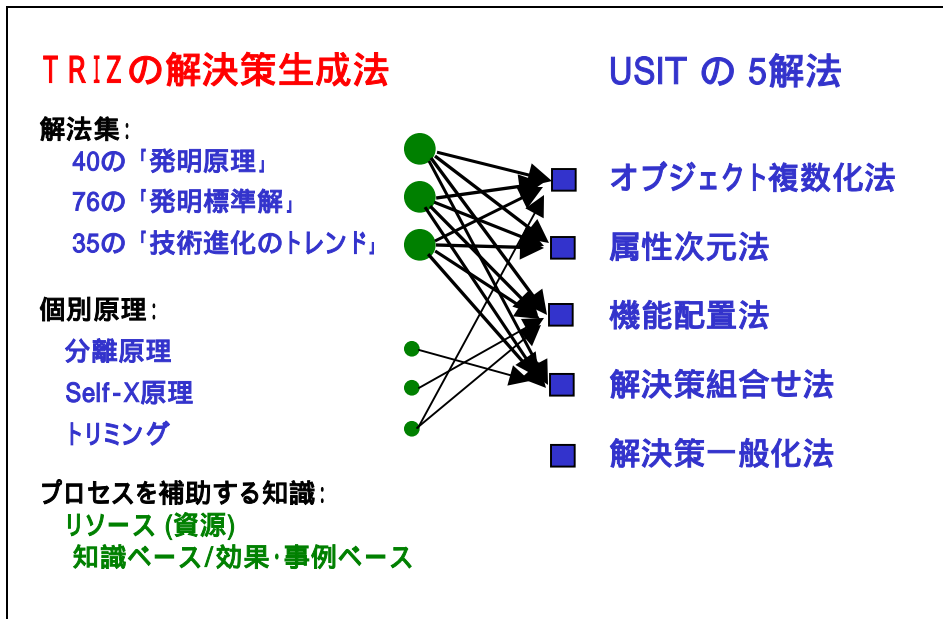
4. T R I Zの全解法を整理して構成した「U S I T解決策生成技法」の体系

U S I Tの解決策生成技法の骨子は以上に述べたように形成されてきていたが、
2002年後半に、中川らはこれをさらに大幅に改良した。

出典: 中川 徹, 古謝秀明, 三原祐治: 「T R I Zの解決策生成諸技法を整理して
U S I Tの5解法に単純化する」, ETRIA2002 国際会議, 2002年11月;
和訳: 『T R I Zホームページ』, 2002年9月掲載。
同: 「U S I Tの解決策生成技法 - T R I Zの解決策生成諸技法を整理して
U S I Tの5解法に単純化した」, 同上の付録。
同: 「U S I T解決策生成法の使い方 -- T R I Zを簡易化・統合化したシス
テム」, TRIZCON2003 投稿論文; 和訳: 『T R I Zホームページ』, 2003年
1月掲載。

T R I Zのさまざまな解法のうち、解決策生成段階のものをすべて集めて分析して、
U S I Tの解決策生成技法 (いままで説明した5種) に振り分け、
それらをU S I Tの枠組みの中で再整理して、階層的な体系に作った。

扱ったT R I Zの解法とそれらのU S I Tへの振り分けを図に示す。



得られた「USITの解決策生成技法の体系」の一覧を以下に示す。

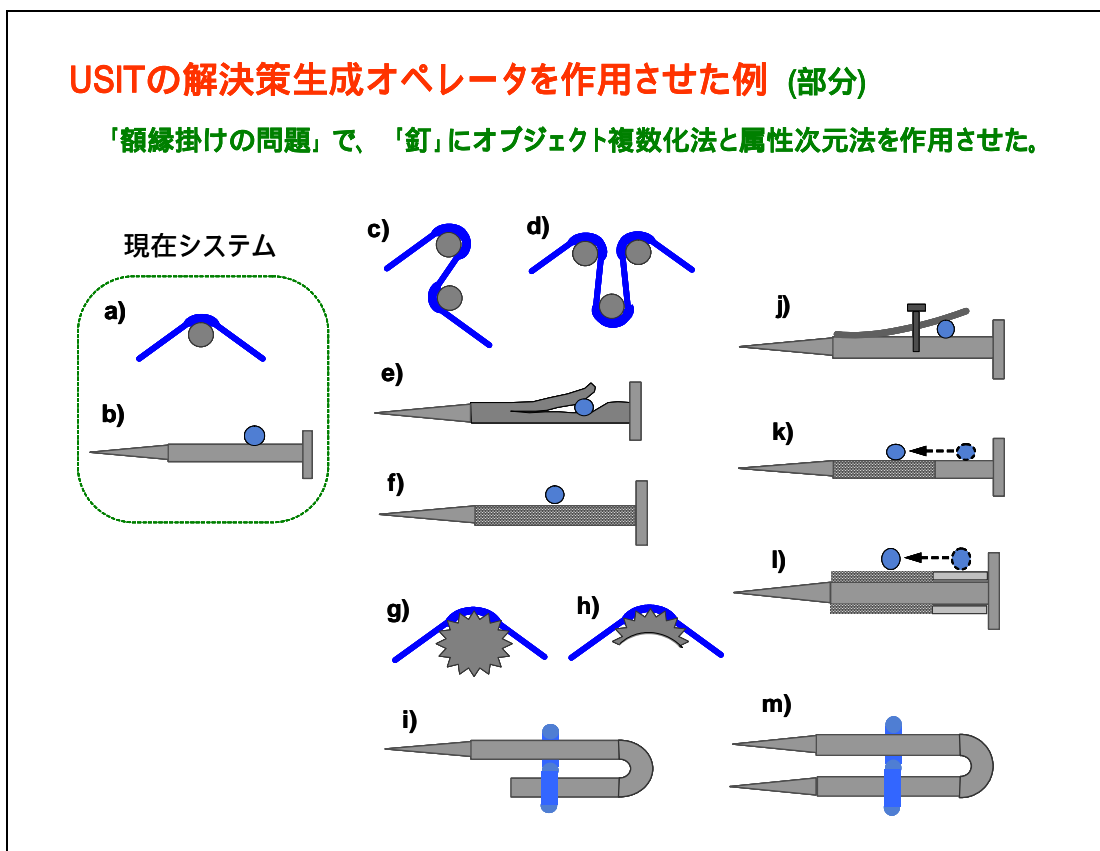
まず、備忘メモ用/掲示用として使う1ページの簡略な一覧表を示す。

USIT 解決策生成法 一覧表	
<p><b style="color: red;">1) オブジェクト複数化法</p> <ul style="list-style-type: none"> a. 消去する b. 多数 (2, 3, ..., 個) に c. 分割 (1/2, 1/3, ... 1/ ずつ) d. 複数をまとめて一つに e. 新規導入/変容 f. 環境から導入 g. 固体から, 粉体, 液体, 気体 へ <p><b style="color: red;">2) 属性次元法</p> <ul style="list-style-type: none"> a. 有害属性を使わない b. 有用な属性を使う c. 有用を強調, 有害を抑制 d. 空間属性を導入, 属性(値)を空間変化 e. 時間属性を導入, 属性(値)を時間変化 f. 相を変える, 内部構造を変える g. ミクロレベルの属性 h. システム全体の性質・機能 	<p><b style="color: red;">3) 機能配置法</p> <ul style="list-style-type: none"> a. 機能を別オブジェクトに b. 複合機能を分割, 分担 c. 二つの機能を統合 d. 新機能を導入 e. 機能を空間的变化, 移動/振動 f. 機能を時間的に変化 g. 検出・測定 of 機能 h. 適応・調整・制御 of 機能 i. 別の物理原理で <p><b style="color: red;">4) 解決策組み合わせ法</p> <ul style="list-style-type: none"> a. 機能的に 組み合わせる b. 空間的に c. 時間的に d. 構造的に e. 原理レベルで f. スーパーシステムに移行 <p><b style="color: red;">5) 解決策一般化法</p> <ul style="list-style-type: none"> a. 用語の一般化と具体化 b. 解決策の階層的な体系

すなわち、USITの5解法はその基本的な定義を崩すことなく、合計32のサブ解法を持ち、各サブ解法には使い方の指針が示され、さらに細部に渡った解法の記述がある。

以下に、「額縁掛けの問題」に適用した例を示し、その使い方を理解しよう。

まず、額縁掛けの「釘」に対して、USIT オペレータを作用させた沢山の例を示す。

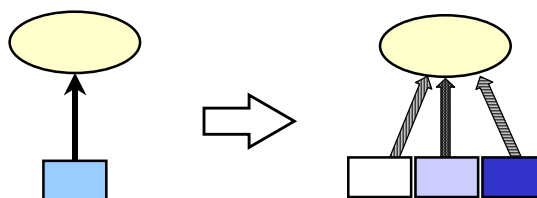


これらを導出するのに使った (使える) USIT オペレータの説明は以下のようである。

(1) オブジェクト複数化法

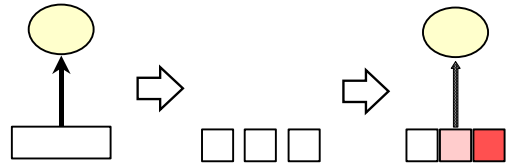
(1b) そのオブジェクトを、多数 (2, 3, ..., 個) にする。

現在のオブジェクトを複数に増やし、それらのオブジェクトの性質を (少しずつ、異なるように) 変容させて一緒に使う。



(1c) そのオブジェクトを、分割 ($1/2, 1/3, \dots, 1/n$ ずつに) する。

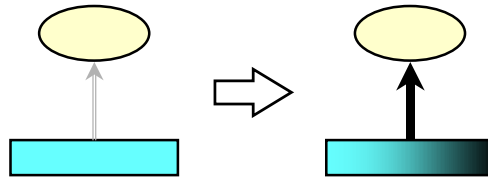
指針: 現在のオブジェクトを複数の部分に ($1/2, 1/3, \dots, 1/n$ ずつに) 分割し、分割した部分部分に (少しずつ、互いに異なる) 変更を加えて、再統合して一緒に用いる。



(2) 属性次元法

(2d) 空間に関する属性を導入・拡張し、また、
 (有害/有用な) 属性およびその値を、空間的に配置/変化させる。

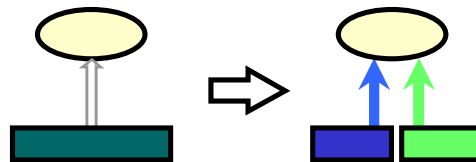
指針: 空間に関する属性でいままで使われていなかったものを導入し
 (あるいは使われていたものを拡張し)、また、空間での位置 (あるいは
 オブジェクトの部分) に応じて、異なる属性を使ったり、
 同じ属性の異なる値を使ったりする。



(3) 機能配置法

(3b) 複合した機能 (複数の機能) を分割して、別のオブジェクトに担わせる。

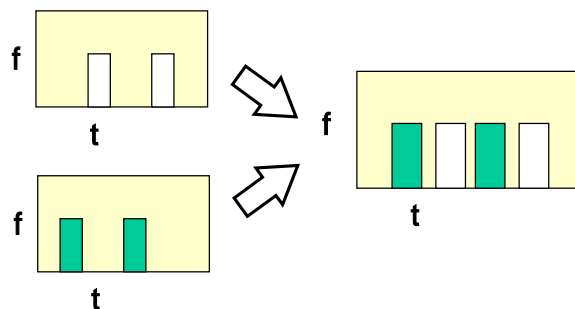
指針: 既存の複合した機能を分割して、別の (既存/新規導入の) オブジェクト
 (またはその部分) に移す。



(4) 解決策組合せ法

(4c) 時間的に組み合わせる。

指針: 複数の解決策を、それを適用する時間に関して組み合わせる。
 順次に、事前に、同時に並行して、事後に、逆順に、交互に、
 パルスの的に、周期的に、間歇的に、条件に応じて割り込み/切り換えて、
 など。



このような例をいろいろ見ていると、段々自分でも考えつけるようになる。

演習 1: 「額縁掛けを傾きにくくする方法」について、自分で思いついたものをなんでも良いから 3 つ書いてみよ。

演習 2: いま書いた自分の案は、USIT の解決策生成法のどれを使ったといえるか?

USIT オペレータを意図的に適用するにはつぎのように考える。

- (1) システムの中の (任意の) オブジェクト一つを取り上げて、それに対して、「オブジェクト複数化法」の(任意の) サブ解法一つを適用することを考える。
- (2) サブ解法の指針に従って、その操作 (例えば、分割する) をともかく行ってみる。
- (3) その結果が問題解決に有効 (かもしれない)と思われるように操作法を調整する。

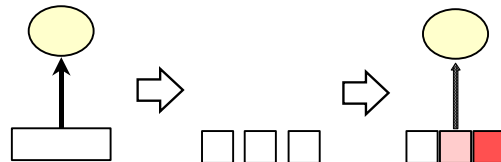
同様の考え方を、属性について、機能について、また解決策の組合せについて、さらに一つ一つの解決策の一般化について考える。

USIT オペレータの適用法の実施例:

例: (1) オブジェクト複数化法

(1c) そのオブジェクトを、分割 (1/2, 1/3, ..., 1/ n ずつに) する。

指針: 現在のオブジェクトを複数の部分に(1/2, 1/3, ..., 1/ n ずつに) 分割し、分割した部分部分に (少しずつ, 互いに異なる)変更を加えて、再統合して一緒に用いる。



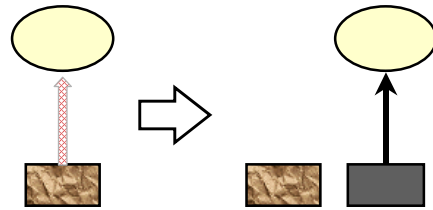
適用結果の例:

釘オブジェクトに適用:	(1) 釘の半分を調節のため滑らかに、半分を保持のため粗く。 (2) 釘を縦に分割してスリットを作り、調節後に紐を差し込む。
紐オブジェクトに適用:	(1) 縋り紐 (三つ編み紐) を使い、縋った紐の間に釘を入れる。 (2) 紐を半分に切り、長さを調節してそれぞれ釘に縛りつける。
フック・オブジェクトに適用:	(1) フックの上部を横に調節可能にし、調節後ネジ止めする。
額縁オブジェクトに適用:	(1) 額縁の一部を背面で左右に動かし、重心位置を調節する。

(3) 機能配置法

(3a) ある機能を、別のオブジェクトに担わせる。

指針： 既存の機能を、より適切な別のオブジェクト（既存または新規導入のもの）に移す。



適用結果の例:

釘の機能に適用:	(1) 釘の調節機能をフックに。 釘に2本の紐をつけ、紐の長さをフックが調節。 (2) 釘の調節機能を紐に。 2本の紐を使い、紐の長さを紐が調節。(バンド止めなど)。
紐の機能に適用:	(1) 紐のぶら下げ機能をフックに。 2つのフックを2本の釘に直接ぶら下げ、 フックの長さを調節して額縁の傾きを調節。
フックの機能に適用:	(1) フックの荷重支持機能を、額縁に担わせる。 額縁の裏面が壁にくっつき、重さを支える。
額縁の機能に適用:	(1) 額縁の傾きを示す機能を新しいオブジェクトに。 額縁に水準器をつけ、傾きを精密に表示させる。

演習： 額縁掛けの問題で、「紐」に対してさまざまな改良を試みよ。アイデアを5つ書け。
オブジェクト複数化法、属性次元法、機能配置法を試みる。

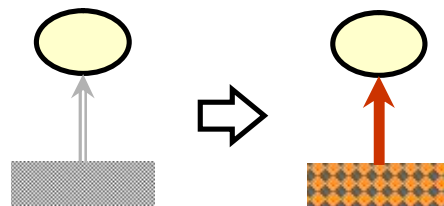
- (1b) 紐を複数にする
- (1c) 紐を分割する
- (2a) 有害属性を使わない、 (2b) 有用属性を使う
- (2d) 空間的属性を導入する、属性を空間的に変化させる
- (2e) 時間的属性を導入する、属性を時間によって変化させる
- (3d) 新しい機能を導入する など

USIT オペレータには、より高度なサブ解法もある。

例： (2)属性次元法

(2g) ミクロのレベルの属性・性質を使う

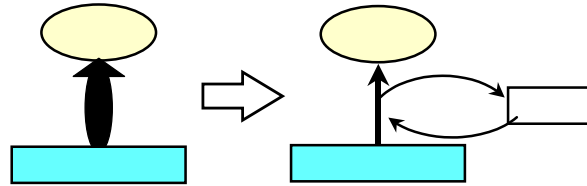
指針： マイクロメートルスケール、ナノメートルスケールでの構造・性質・相互作用を考慮して、原理レベルから考えて問題を解決する。



(3) 機能配置法

(3h) 適応・調整・制御の機能を導入/拡張する

指針： システムに適応・調整・制御などの機能を導入・拡張し、システムを高度化・知能化する。



このようにして、多数のアイデアが出てくるが、それを整理して考えると次第に解決策の体系が顕れて来る。(紙の上でも、頭の中でも。)

アイデア生成後のプロセス:

- (1) アイデア (の断片) をできるだけ多く考え出す。
- (2) それらをさらに一般化し、具体化して、膨らませていく。
- (3) またそれらをいろいろに組み合わせて、さらに発展させる。
- (4) アイデアを似た観点のものでグループ化する。
- (5) アイデアの体系を作っていく。
- (6) アイデアのエッセンスが何かを考える。
- (7) どのアイデアが最も本質的に問題を解決しているのかを考える。
- (8) 問題の本質的な矛盾とその解決策を明確にする。
- (9) 解決策を具体的にして、解決策の設計案を作る。
- (10) 解決策の設計案に問題点があるなら、その克服策 (解決策) を考える。
- (11) 得られたいくつかの解決策を試作・実験して評価する。
- (12) すぐれた解決策を実際の製品にまで発展させていく。

ここで、上記(7) の点を例にして考えよう。

(7) 「どのアイデアが最も本質的に問題を解決しているのか?」

具体的には、前述の「粗い表面と滑らかな表面をもつ釘」を例にする。



この解決策に対する下記の「解釈」(エッセンスの理解) のうち、どれが最も本質的か?

- (a) オブジェクト複数化法: 「釘」オブジェクトを半分ずつにして、性質を変えて統合。
- (b) 属性次元法: 釘の表面の「滑らかさ」の属性の値を、釘の部分によって変えた。
- (c) 機能配置法: 釘の「調節」機能と「保持」機能を分離し、釘の部分毎に担当させた。
- (d) 解決策組合せ法: 釘を滑らかにして調節しやすくする解決策と、
釘の表面を粗くして、傾きにくくさせる解決策とを、
釘の部分分割することにより組み合わせた。
- (e) 解決策組合せ法 (その 2): 上記の二つの解決策を、時間によって組み合わせた。

最も本質的なのは、「時間に関する組合せ」という解釈である。

額縁掛けのシステムは、調節しているときには「滑らかに調節できる」こと。
調節後は、「しっかりと固定されている」こと。

調節しているときには、スムーズに動いて調整でき、
調整後保持するときには、しっかりと固定できているとよい。

このように考えると、調節と固定の具体的な方法がどんどん思い浮かんで来る。

演習: このような「調節し」それから「固定」する仕組みを列挙せよ。

演習: 上記の「調節し、その後固定する」仕組みを、「額縁掛けの問題」に
適用して、解決策を 10 件書き出してみよ。

このように、調節時と調節後の「時間による分離」の解決策が本質。
これは TRIZ でいう「物理的矛盾」を確認したこと。
その「物理的矛盾」を解決した解決策がもっとも本質的になる。

USIT の技法を使いながら、TRIZ の思想・考え方を活用するとよい。